



This book is provided in digital form with the permission of the rightsholder as part of a Google project to make the world's books discoverable online.

The rightsholder has graciously given you the freedom to download all pages of this book. No additional commercial or other uses have been granted.

Please note that all copyrights remain reserved.

About Google Books

Google's mission is to organize the world's information and to make it universally accessible and useful. Google Books helps readers discover the world's books while helping authors and publishers reach new audiences. You can search through the full text of this book on the web at <http://books.google.com/>

Big Data Now

Current Perspectives from O'Reilly Radar



O'REILLY®

O'REILLY®
Strata
Making Data Work

Big Data Now

Edited by

O'Reilly Radar Team

Editor

Allen Noren

Copyright © 2011

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Big Data Now* and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

O'REILLY®

O'Reilly Media

Foreword

This collection represents the full spectrum of data-related content we've published on O'Reilly Radar over the last year. Mike Loukides kicked things off in June 2010 with "What is data science?" and from there we've pursued the various threads and themes that naturally emerged. Now, roughly a year later, we can look back over all we've covered and identify a number of core data areas:

Chapter 1—The tools and technologies that drive data science are of course essential to this space, but the varied techniques being applied are also key to understanding the big data arena.

Chapter 2—The opportunities and ambiguities of the data space are evident in discussions around privacy, the implications of data-centric industries, and the debate about the phrase "data science" itself.

Chapter 3—A "data product" can emerge from virtually any domain, including everything from data startups to established enterprises to media/journalism to education and research.

Chapter 4—Take a closer look at the actions connected to data—the finding, organizing, and analyzing that provide organizations of all sizes with the information they need to compete.

To be clear: This is the story up to this point. In the weeks and months ahead we'll certainly see important shifts in the data landscape. We'll continue to chronicle this space through ongoing Radar coverage and our series of online and in-person Strata events. We hope you'll join us.

—Mac Slocum

Managing Editor, O'Reilly Radar

Chapter 1. Data Science and Data Tools

What is data science?

The future belongs to the companies and people that turn data into products.

by Mike Loukides

We've all heard it: according to Hal Varian, statistics is the next sexy job. Five years ago, in *What is Web 2.0*, Tim O'Reilly said that "data is the next Intel Inside." But what does that statement mean? Why do we suddenly care about statistics and about data?

In this post, I examine the many sides of data science—the technologies, the companies and the unique skill sets.

What is data science?

The web is full of "data-driven apps." Almost any e-commerce application is a data-driven application. There's a database behind a web front end, and middleware that talks to a number of other databases and data services (credit card processing companies, banks, and so on). But merely using data isn't really what we mean by "data science." A data application acquires its value from the data itself, and creates more data as a result. It's not just an application with data; it's a data product. Data science enables the creation of data products.

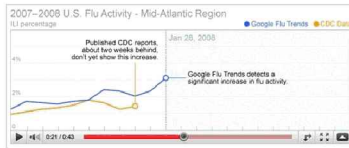
One of the earlier data products on the Web was the CDDB database. The developers of CDDB realized that any CD had a unique signature, based on the exact length (in samples) of each track on the CD. Gracenote built a database of track lengths, and coupled it to a database of album metadata (track titles, artists, album titles). If you've ever used iTunes to rip a CD, you've taken advantage of this database. Before it does anything else, iTunes reads the length of every track, sends it to

CDDB, and gets back the track titles. If you have a CD that's not in the database (including a CD you've made yourself), you can create an entry for an unknown album. While this sounds simple enough, it's revolutionary: CDDB views music as data, not as audio, and creates new value in doing so. Their business is fundamentally different from selling music, sharing music, or analyzing musical tastes (though these can also be "data products"). CDDB arises entirely from viewing a musical problem as a data problem.

Google is a master at creating data products. Here's a few examples:

- Google's breakthrough was realizing that a search engine could use input other than the text on the page. Google's PageRank algorithm was among the first to use data outside of the page itself, in particular, the number of links pointing to a page. Tracking links made Google searches much more useful, and PageRank has been a key ingredient to the company's success.
- Spell checking isn't a terribly difficult problem, but by suggesting corrections to misspelled searches, and observing what the user clicks in response, Google made it much more accurate. They've built a dictionary of common misspellings, their corrections, and the contexts in which they occur.
- Speech recognition has always been a hard problem, and it remains difficult. But Google has made huge strides by using the voice data they've collected, and has been able to integrate voice search into their core search engine.
- During the Swine Flu epidemic of 2009, Google was able to track the progress of the epidemic by following searches for flu-related topics.

FLU TRENDS



Google was able to spot trends in the Swine Flu epidemic roughly two weeks before the Center for Disease Control by analyzing searches that people were making in different regions of the country.

Google isn't the only company that knows how to use data. Facebook and LinkedIn use patterns of friendship relationships to suggest other people you may know, or should know, with sometimes frightening accuracy. Amazon saves your searches, correlates what you search for with what other users search for, and uses it to create surprisingly appropriate recommendations. These recommendations are "data products" that help to drive Amazon's more traditional retail business. They come about because Amazon understands that a book isn't just a book, a camera isn't just a camera, and a customer isn't just a customer; customers generate a trail of "data exhaust" that can be mined and put to use, and a camera is a cloud of data that can be correlated with the customers' behavior, the data they leave every time they visit the site.

The thread that ties most of these applications together is that data collected from users provides added value. Whether that data is search terms, voice samples, or product reviews, the users are in a feedback loop in which they contribute to the products they use. That's the beginning of data science.

In the last few years, there has been an explosion in the amount of data that's available. Whether we're talking about web server logs, tweet streams, online transaction records, "citizen science," data from sensors, government data, or some other source, the problem isn't finding data, it's figuring out what to do with it. And it's not just companies using their own data, or the data contributed by their users. It's increasingly common to mashup data from a number of sources. "Data Mashups in R"

analyzes mortgage foreclosures in Philadelphia County by taking a public report from the county sheriff's office, extracting addresses and using Yahoo to convert the addresses to latitude and longitude, then using the geographical data to place the foreclosures on a map (another data source), and group them by neighborhood, valuation, neighborhood per-capita income, and other socio-economic factors.

The question facing every company today, every startup, every non-profit, every project site that wants to attract a community, is how to use data effectively—not just their own data, but all the data that's available and relevant. Using data effectively requires something different from traditional statistics, where actuaries in business suits perform arcane but fairly well-defined kinds of analysis. What differentiates data science from statistics is that data science is a holistic approach. We're increasingly finding data in the wild, and data scientists are involved with gathering data, massaging it into a tractable form, making it tell its story, and presenting that story to others.

To get a sense for what skills are required, let's look at the data lifecycle: where it comes from, how you use it, and where it goes.

Where data comes from

Data is everywhere: your government, your web server, your business partners, even your body. While we aren't drowning in a sea of data, we're finding that almost everything can (or has) been instrumented. At O'Reilly, we frequently combine publishing industry data from Nielsen BookScan with our own sales data, publicly available Amazon data, and even job data to see what's happening in the publishing industry. Sites like Infochimps and Factual provide access to many large datasets, including climate data, MySpace activity streams, and game logs from sporting events. Factual enlists users to update and improve its datasets, which cover topics as diverse as endocrinologists to hiking trails.

Much of the data we currently work with is the direct consequence of Web 2.0, and of Moore's Law applied to data. The web has people spending more time online, and leaving a trail of data wherever they go.

Mobile applications leave an even richer data trail, since many of them are annotated with geolocation, or involve video or audio, all of which can be mined. Point-of-sale devices and frequent-shopper's cards make it possible to capture all of your retail transactions, not just the ones you make online. All of this data would be useless if we couldn't store it, and that's where Moore's Law comes in. Since the early '80s, processor speed has increased from 10 MHz to 3.6 GHz—an increase of 360 (not counting increases in word length and number of cores). But we've seen much bigger increases in storage capacity, on every level. RAM has moved from \$1,000/MB to roughly \$25/GB—a price reduction of about 40000, to say nothing of the reduction in size and increase in speed. Hitachi made the first gigabyte disk drives in 1982, weighing in at roughly 250 pounds; now terabyte drives are consumer equipment, and a 32 GB microSD card weighs about half a gram. Whether you look at bits per gram, bits per dollar, or raw capacity, storage has more than kept pace with the increase of CPU speed.

1956 DISK DRIVE



One of the first commercial disk drives from IBM. It has a 5 MB capacity and it's stored in a cabinet roughly the size of a luxury refrigerator. In contrast, a 32 GB microSD card measures around 5/8 x 3/8 inch and weighs about 0.5 gram.

Photo: Mike Loukides. Disk drive on display at IBM Almaden Research

The importance of Moore's law as applied to data isn't just geek pyrotechnics. Data expands to fill the space you have to store it. The more storage is available, the more data you will find to put into it. The data exhaust you leave behind whenever you surf the web, friend someone on Facebook, or make a purchase in your local supermarket, is

all carefully collected and analyzed. Increased storage capacity demands increased sophistication in the analysis and use of that data. That's the foundation of data science.

So, how do we make that data useful? The first step of any data analysis project is "data conditioning," or getting data into a state where it's usable. We are seeing more data in formats that are easier to consume: Atom data feeds, web services, microformats, and other newer technologies provide data in formats that's directly machine-consumable. But old-style screen scraping hasn't died, and isn't going to die. Many sources of "wild data" are extremely messy. They aren't well-behaved XML files with all the metadata nicely in place. The foreclosure data used in "Data Mashups in R" was posted on a public website by the Philadelphia county sheriff's office. This data was presented as an HTML file that was probably generated automatically from a spreadsheet. If you've ever seen the HTML that's generated by Excel, you know that's going to be fun to process.

Data conditioning can involve cleaning up messy HTML with tools like Beautiful Soup, natural language processing to parse plain text in English and other languages, or even getting humans to do the dirty work. You're likely to be dealing with an array of data sources, all in different forms. It would be nice if there was a standard set of tools to do the job, but there isn't. To do data conditioning, you have to be ready for whatever comes, and be willing to use anything from ancient Unix utilities such as awk to XML parsers and machine learning libraries. Scripting languages, such as Perl and Python, are essential.

Once you've parsed the data, you can start thinking about the quality of your data. Data is frequently missing or incongruous. If data is missing, do you simply ignore the missing points? That isn't always possible. If data is incongruous, do you decide that something is wrong with badly behaved data (after all, equipment fails), or that the incongruous data is telling its own story, which may be more interesting? It's reported that the discovery of ozone layer depletion was delayed because automated data collection tools discarded readings that were too low^[1]. In data science, what you have is frequently all you're going to get. It's usually

impossible to get “better” data, and you have no alternative but to work with the data at hand.

If the problem involves human language, understanding the data adds another dimension to the problem. Roger Magoulas, who runs the data analysis group at O’Reilly, was recently searching a database for Apple job listings requiring geolocation skills. While that sounds like a simple task, the trick was disambiguating “Apple” from many job postings in the growing Apple industry. To do it well you need to understand the grammatical structure of a job posting; you need to be able to parse the English. And that problem is showing up more and more frequently. Try using Google Trends to figure out what’s happening with the Cassandra database or the Python language, and you’ll get a sense of the problem. Google has indexed many, many websites about large snakes. Disambiguation is never an easy task, but tools like the Natural Language Toolkit library can make it simpler.

When natural language processing fails, you can replace artificial intelligence with human intelligence. That’s where services like Amazon’s Mechanical Turk come in. If you can split your task up into a large number of subtasks that are easily described, you can use Mechanical Turk’s marketplace for cheap labor. For example, if you’re looking at job listings, and want to know which originated with Apple, you can have real people do the classification for roughly \$0.01 each. If you have already reduced the set to 10,000 postings with the word “Apple,” paying humans \$0.01 to classify them only costs \$100.

Working with data at scale

We’ve all heard a lot about “big data,” but “big” is really a red herring. Oil companies, telecommunications companies, and other data-centric industries have had huge datasets for a long time. And as storage capacity continues to expand, today’s “big” is certainly tomorrow’s “medium” and next week’s “small.” The most meaningful definition I’ve heard: *“big data” is when the size of the data itself becomes part of the problem.* We’re discussing data problems ranging from gigabytes to

petabytes of data. At some point, traditional techniques for working with data run out of steam.

What are we trying to do with data that's different? According to Jeff Hammerbacher^[2] (@hackingdata), we're trying to build information platforms or dataspace. Information platforms are similar to traditional data warehouses, but different. They expose rich APIs, and are designed for exploring and understanding the data rather than for traditional analysis and reporting. They accept all data formats, including the most messy, and their schemas evolve as the understanding of the data changes.

Most of the organizations that have built data platforms have found it necessary to go beyond the relational database model. Traditional relational database systems stop being effective at this scale. Managing sharding and replication across a horde of database servers is difficult and slow. The need to define a schema in advance conflicts with reality of multiple, unstructured data sources, in which you may not know what's important until after you've analyzed the data. Relational databases are designed for consistency, to support complex transactions that can easily be rolled back if any one of a complex set of operations fails. While rock-solid consistency is crucial to many applications, it's not really necessary for the kind of analysis we're discussing here. Do you really care if you have 1,010 or 1,012 Twitter followers? Precision has an allure, but in most data-driven applications outside of finance, that allure is deceptive. Most data analysis is comparative: if you're asking whether sales to Northern Europe are increasing faster than sales to Southern Europe, you aren't concerned about the difference between 5.92 percent annual growth and 5.93 percent.

To store huge datasets effectively, we've seen a new breed of databases appear. These are frequently called NoSQL databases, or Non-Relational databases, though neither term is very useful. They group together fundamentally dissimilar products by telling you what they aren't. Many of these databases are the logical descendants of Google's BigTable and Amazon's Dynamo, and are designed to be distributed across many nodes, to provide "eventual consistency" but not absolute consistency,

and to have very flexible schema. While there are two dozen or so products available (almost all of them open source), a few leaders have established themselves:

- **Cassandra:** Developed at Facebook, in production use at Twitter, Rackspace, Reddit, and other large sites. Cassandra is designed for high performance, reliability, and automatic replication. It has a very flexible data model. A new startup, Riptano, provides commercial support.
- **HBase:** Part of the Apache Hadoop project, and modelled on Google's BigTable. Suitable for extremely large databases (billions of rows, millions of columns), distributed across thousands of nodes. Along with Hadoop, commercial support is provided by Cloudera.

Storing data is only part of building a data platform, though. Data is only useful if you can do something with it, and enormous datasets present computational problems. Google popularized the MapReduce approach, which is basically a divide-and-conquer strategy for distributing an extremely large problem across an extremely large computing cluster. In the "map" stage, a programming task is divided into a number of identical subtasks, which are then distributed across many processors; the intermediate results are then combined by a single reduce task. In hindsight, MapReduce seems like an obvious solution to Google's biggest problem, creating large searches. It's easy to distribute a search across thousands of processors, and then combine the results into a single set of answers. What's less obvious is that MapReduce has proven to be widely applicable to many large data problems, ranging from search to machine learning.

The most popular open source implementation of MapReduce is the Hadoop project. Yahoo's claim that they had built the world's largest production Hadoop application, with 10,000 cores running Linux, brought it onto center stage. Many of the key Hadoop developers have found a home at Cloudera, which provides commercial support. Amazon's Elastic MapReduce makes it much easier to put Hadoop to work without investing in racks of Linux machines, by providing

preconfigured Hadoop images for its EC2 clusters. You can allocate and de-allocate processors as needed, paying only for the time you use them.

Hadoop goes far beyond a simple MapReduce implementation (of which there are several); it's the key component of a data platform. It incorporates HDFS, a distributed filesystem designed for the performance and reliability requirements of huge datasets; the HBase database; Hive, which lets developers explore Hadoop datasets using SQL-like queries; a high-level dataflow language called Pig; and other components. If anything can be called a one-stop information platform, Hadoop is it.

Hadoop has been instrumental in enabling “agile” data analysis. In software development, “agile practices” are associated with faster product cycles, closer interaction between developers and consumers, and testing. Traditional data analysis has been hampered by extremely long turn-around times. If you start a calculation, it might not finish for hours, or even days. But Hadoop (and particularly Elastic MapReduce) make it easy to build clusters that can perform computations on long datasets quickly. Faster computations make it easier to test different assumptions, different datasets, and different algorithms. It's easier to consult with clients to figure out whether you're asking the right questions, and it's possible to pursue intriguing possibilities that you'd otherwise have to drop for lack of time.

Hadoop is essentially a batch system, but Hadoop Online Prototype (HOP) is an experimental project that enables stream processing. Hadoop processes data as it arrives, and delivers intermediate results in (near) real-time. Near real-time data analysis enables features like trending topics on sites like Twitter. These features only require soft real-time; reports on trending topics don't require millisecond accuracy. As with the number of followers on Twitter, a “trending topics” report only needs to be current to within five minutes—or even an hour. According to Hilary Mason (@hmason), data scientist at bit.ly, it's possible to precompute much of the calculation, then use one of the experiments in real-time MapReduce to get presentable results.

Machine learning is another essential tool for the data scientist. We now expect web and mobile applications to incorporate recommendation engines, and building a recommendation engine is a quintessential artificial intelligence problem. You don't have to look at many modern web applications to see classification, error detection, image matching (behind Google Goggles and SnapTell) and even face detection—an ill-advised mobile application lets you take someone's picture with a cell phone, and look up that person's identity using photos available online. Andrew Ng's Machine Learning course is one of the most popular courses in computer science at Stanford, with hundreds of students (this video is highly recommended).

There are many libraries available for machine learning: PyBrain in Python, Elefant, Weka in Java, and Mahout (coupled to Hadoop). Google has just announced their Prediction API, which exposes their machine learning algorithms for public use via a RESTful interface. For computer vision, the OpenCV library is a de-facto standard.

Mechanical Turk is also an important part of the toolbox. Machine learning almost always requires a "training set," or a significant body of known data with which to develop and tune the application. The Turk is an excellent way to develop training sets. Once you've collected your training data (perhaps a large collection of public photos from Twitter), you can have humans classify them inexpensively—possibly sorting them into categories, possibly drawing circles around faces, cars, or whatever interests you. It's an excellent way to classify a few thousand data points at a cost of a few cents each. Even a relatively large job only costs a few hundred dollars.

While I haven't stressed traditional statistics, building statistical models plays an important role in any data analysis. According to Mike Driscoll (@dataspora), statistics is the "grammar of data science." It is crucial to "making data speak coherently." We've all heard the joke that eating pickles causes death, because everyone who dies has eaten pickles. That joke doesn't work if you understand what correlation means. More to the point, it's easy to notice that one advertisement for *R in a Nutshell* generated 2 percent more conversions than another. But it takes statistics

to know whether this difference is significant, or just a random fluctuation. Data science isn't just about the existence of data, or making guesses about what that data might mean; it's about testing hypotheses and making sure that the conclusions you're drawing from the data are valid. Statistics plays a role in everything from traditional business intelligence (BI) to understanding how Google's ad auctions work. Statistics has become a basic skill. It isn't superseded by newer techniques from machine learning and other disciplines; it complements them.

While there are many commercial statistical packages, the open source R language—and its comprehensive package library, CRAN—is an essential tool. Although R is an odd and quirky language, particularly to someone with a background in computer science, it comes close to providing “one stop shopping” for most statistical work. It has excellent graphics facilities; CRAN includes parsers for many kinds of data; and newer extensions extend R into distributed computing. If there's a single tool that provides an end-to-end solution for statistics work, R is it.

Making data tell its story

A picture may or may not be worth a thousand words, but a picture is certainly worth a thousand numbers. The problem with most data analysis algorithms is that they generate a set of numbers. To understand what the numbers mean, the stories they are really telling, you need to generate a graph. Edward Tufte's *Visual Display of Quantitative Information* is the classic for data visualization, and a foundational text for anyone practicing data science. But that's not really what concerns us here. Visualization is crucial to each stage of the data scientist.

According to Martin Wattenberg (@wattenberg, founder of Flowing Media), visualization is key to data conditioning: if you want to find out just how bad your data is, try plotting it. Visualization is also frequently the first step in analysis. Hilary Mason says that when she gets a new data set, she starts by making a dozen or more scatter plots, trying to get a sense of what might be interesting. Once you've gotten some hints at

what the data might be saying, you can follow it up with more detailed analysis.

There are many packages for plotting and presenting data. GnuPlot is very effective; R incorporates a fairly comprehensive graphics package; Casey Reas' and Ben Fry's Processing is the state of the art, particularly if you need to create animations that show how things change over time. At IBM's Many Eyes, many of the visualizations are full-fledged interactive applications.

Nathan Yau's FlowingData blog is a great place to look for creative visualizations. One of my favorites is this animation of the growth of Walmart over time. And this is one place where "art" comes in: not just the aesthetics of the visualization itself, but how you understand it. Does it look like the spread of cancer throughout a body? Or the spread of a flu virus through a population? Making data tell its story isn't just a matter of presenting results; it involves making connections, then going back to other data sources to verify them. Does a successful retail chain spread like an epidemic, and if so, does that give us new insights into how economies work? That's not a question we could even have asked a few years ago. There was insufficient computing power, the data was all locked up in proprietary sources, and the tools for working with the data were insufficient. It's the kind of question we now ask routinely.

Data scientists

Data science requires skills ranging from traditional computer science to mathematics to art. Describing the data science group he put together at Facebook (possibly the first data science group at a consumer-oriented web property), Jeff Hammerbacher said:

... on any given day, a team member could author a multistage processing pipeline in Python, design a hypothesis test, perform a regression analysis over data samples with R, design and implement an algorithm for some data-intensive product or service in Hadoop, or communicate the results of our analyses to other members of the organization^[3]

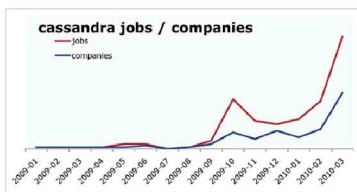
Where do you find the people this versatile? According to DJ Patil, chief scientist at LinkedIn (@dpatil), the best data scientists tend to be “hard scientists,” particularly physicists, rather than computer science majors. Physicists have a strong mathematical background, computing skills, and come from a discipline in which survival depends on getting the most from the data. They have to think about the big picture, the big problem. When you’ve just spent a lot of grant money generating data, you can’t just throw the data out if it isn’t as clean as you’d like. You have to make it tell its story. You need some creativity for when the story the data is telling isn’t what you think it’s telling.

Scientists also know how to break large problems up into smaller problems. Patil described the process of creating the group recommendation feature at LinkedIn. It would have been easy to turn this into a high-ceremony development project that would take thousands of hours of developer time, plus thousands of hours of computing time to do massive correlations across LinkedIn’s membership. But the process worked quite differently: it started out with a relatively small, simple program that looked at members’ profiles and made recommendations accordingly. Asking things like, did you go to Cornell? Then you might like to join the Cornell Alumni group. It then branched out incrementally. In addition to looking at profiles, LinkedIn’s data scientists started looking at events that members attended. Then at books members had in their libraries. The result was a valuable data product that analyzed a huge database—but it was never conceived as such. It started small, and added value iteratively. It was an agile, flexible process that built toward its goal incrementally, rather than tackling a huge mountain of data all at once.

This is the heart of what Patil calls “data jiu-jitsu”—using smaller auxiliary problems to solve a large, difficult problem that appears intractable. CDDDB is a great example of data jiu-jitsu: identifying music by analyzing an audio stream directly is a very difficult problem (though not unsolvable—see midomi, for example). But the CDDDB staff used data creatively to solve a much more tractable problem that gave them

the same result. Computing a signature based on track lengths, and then looking up that signature in a database, is trivially simple.

HIRING TRENDS FOR DATA SCIENCE



It's not easy to get a handle on jobs in data science. However, data from [O'Reilly Research](#) shows a steady year-over-year increase in Hadoop and Cassandra job listings, which are good proxies for the "data science" market as a whole. This graph shows the increase in Cassandra jobs, and the companies listing Cassandra positions, over time.

Entrepreneurship is another piece of the puzzle. Patil's first flippant answer to "what kind of person are you looking for when you hire a data scientist?" was "someone you would start a company with." That's an important insight: we're entering the era of products that are built on data. We don't yet know what those products are, but we do know that the winners will be the people, and the companies, that find those products. Hilary Mason came to the same conclusion. Her job as scientist at bit.ly is really to investigate the data that bit.ly is generating, and find out how to build interesting products from it. No one in the nascent data industry is trying to build the 2012 Nissan Stanza or Office 2015; they're all trying to find new products. In addition to being physicists, mathematicians, programmers, and artists, they're entrepreneurs.

Data scientists combine entrepreneurship with patience, the willingness to build data products incrementally, the ability to explore, and the ability to iterate over a solution. They are inherently interdisciplinary. They can tackle all aspects of a problem, from initial data collection and data conditioning to drawing conclusions. They can think outside the box to come up with new ways to view the problem, or to work with very

broadly defined problems: “here’s a lot of data, what can you make from it?”

The future belongs to the companies who figure out how to collect and use data successfully. Google, Amazon, Facebook, and LinkedIn have all tapped into their datastreams and made that the core of their success. They were the vanguard, but newer companies like bit.ly are following their path. Whether it’s mining your personal biology, building maps from the shared experience of millions of travellers, or studying the URLs that people pass to others, the next generation of successful businesses will be built around data. The part of Hal Varian’s quote that nobody remembers says it all:

The ability to take data—to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it—that’s going to be a hugely important skill in the next decades.

Data is indeed the new Intel Inside.

[1] The NASA article denies this, but also says that in 1984, they decided that the low values (which went back to the 70s) were “real.” Whether humans or software decided to ignore anomalous data, it appears that data was ignored.

[2] “Information Platforms as Dataspaces,” by Jeff Hammerbacher (in *Beautiful Data*)

[3] “Information Platforms as Dataspaces,” by Jeff Hammerbacher (in *Beautiful Data*)

The SMAQ stack for big data

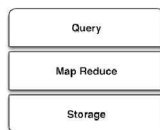
Storage, MapReduce and Query are ushering in data-driven products and services.

by Edd Dumbill

“Big data” is data that becomes large enough that it cannot be processed using conventional methods. Creators of web search engines were among the first to confront this problem. Today, social networks, mobile phones, sensors and science contribute to petabytes of data created daily.

To meet the challenge of processing such large data sets, Google created MapReduce. Google’s work and Yahoo’s creation of the Hadoop MapReduce implementation has spawned an ecosystem of big data processing tools.

As MapReduce has grown in popularity, a stack for big data systems has emerged, comprising layers of Storage, MapReduce and Query (SMAQ). SMAQ systems are typically open source, distributed, and run on commodity hardware.

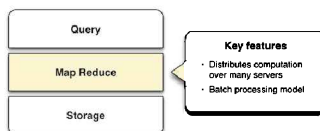


In the same way the commodity LAMP stack of Linux, Apache, MySQL and PHP changed the landscape of web applications, SMAQ systems are bringing commodity big data processing to a broad audience. SMAQ systems underpin a new era of innovative data-driven products and services, in the same way that LAMP was a critical enabler for Web 2.0.

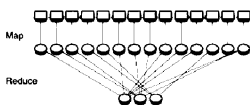
Though dominated by Hadoop-based architectures, SMAQ encompasses a variety of systems, including leading NoSQL databases. This paper describes the SMAQ stack and where today’s big data tools fit into the picture.

MapReduce

Created at Google in response to the problem of creating web search indexes, the MapReduce framework is the powerhouse behind most of today's big data processing. The key innovation of MapReduce is the ability to take a query over a data set, divide it, and run it in parallel over many nodes. This distribution solves the issue of data too large to fit onto a single machine.



To understand how MapReduce works, look at the two phases suggested by its name. In the map phase, input data is processed, item by item, and transformed into an intermediate data set. In the reduce phase, these intermediate results are reduced to a summarized data set, which is the desired end result.



A simple example of MapReduce is the task of counting the number of unique words in a document. In the map phase, each word is identified and given the count of 1. In the reduce phase, the counts are added together for each word.

If that seems like an obscure way of doing a simple task, that's because it is. In order for MapReduce to do its job, the map and reduce phases must obey certain constraints that allow the work to be parallelized.

Translating queries into one or more MapReduce steps is not an intuitive process. Higher-level abstractions have been developed to ease this, discussed under Query below.

An important way in which MapReduce-based systems differ from conventional databases is that they process data in a batch-oriented fashion. Work must be queued for execution, and may take minutes or hours to process.

Using MapReduce to solve problems entails three distinct operations:

- **Loading the data**—This operation is more properly called Extract, Transform, Load (ETL) in data warehousing terminology. Data must be extracted from its source, structured to make it ready for processing, and loaded into the storage layer for MapReduce to operate on it.
- **MapReduce**—This phase will retrieve data from storage, process it, and return the results to the storage.
- **Extracting the result**—Once processing is complete, for the result to be useful to humans, it must be retrieved from the storage and presented.

Many SMAQ systems have features designed to simplify the operation of each of these stages.

Hadoop MapReduce

Hadoop is the dominant open source MapReduce implementation. Funded by Yahoo, it emerged in 2006 and, according to its creator Doug Cutting, reached “web scale” capability in early 2008.

The Hadoop project is now hosted by Apache. It has grown into a large endeavor, with multiple subprojects that together comprise a full SMAQ stack.

Since it is implemented in Java, Hadoop’s MapReduce implementation is accessible from the Java programming language. Creating MapReduce jobs involves writing functions to encapsulate the map and reduce stages of the computation. The data to be processed must be loaded into the Hadoop Distributed Filesystem.

Taking the word-count example from above, a suitable map function might look like the following (taken from the Hadoop MapReduce documentation, the key operations shown in bold).

```
public static class Map
    extends Mapper<LongWritable, Text, Text,
IntWritable> {
```

```

        private final static IntWritable one = new
IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value,
Context context)
            throws IOException, InterruptedException {

            String line = value.toString();
            StringTokenizer tokenizer = new
StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {

word.set(tokenizer.nextToken());context.write(word,
one);

            }
        }
    }
}

```

The corresponding reduce function sums the counts for each word.

```

public static class Reduce
    extends Reducer<Text, IntWritable,
Text, IntWritable> {

    public void reduce(Text key,
Iterable<IntWritable> values,
        Context context) throws IOException,
InterruptedException {

        int sum = 0;
        for (IntWritable val : values) {sum +=
val.get();}context.write(key, new IntWritable(sum));
    }
}

```

The process of running a MapReduce job with Hadoop involves the following steps:

- Defining the MapReduce stages in a Java program

- Loading the data into the filesystem
- Submitting the job for execution
- Retrieving the results from the filesystem

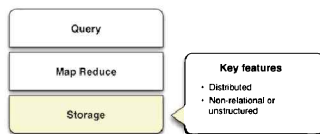
Run via the standalone Java API, Hadoop MapReduce jobs can be complex to create, and necessitate programmer involvement. A broad ecosystem has grown up around Hadoop to make the task of loading and processing data more straightforward.

Other implementations

MapReduce has been implemented in a variety of other programming languages and systems, a list of which may be found in Wikipedia's entry for MapReduce. Notably, several NoSQL database systems have integrated MapReduce, and are described later in this paper.

Storage

MapReduce requires storage from which to fetch data and in which to store the results of the computation. The data expected by MapReduce is not relational data, as used by conventional databases. Instead, data is consumed in chunks, which are then divided among nodes and fed to the map phase as key-value pairs. This data does not require a schema, and may be unstructured. However, the data must be available in a distributed fashion, to serve each processing node.



The design and features of the storage layer are important not just because of the interface with MapReduce, but also because they affect the ease with which data can be loaded and the results of computation extracted and searched.

Hadoop Distributed File System

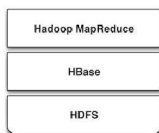
The standard storage mechanism used by Hadoop is the Hadoop Distributed File System, HDFS. A core part of Hadoop, HDFS has the following features, as detailed in the HDFS design document.

- **Fault tolerance**—Assuming that failure will happen allows HDFS to run on commodity hardware.
- **Streaming data access**—HDFS is written with batch processing in mind, and emphasizes high throughput rather than random access to data.
- **Extreme scalability**—HDFS will scale to petabytes; such an installation is in production use at Facebook.
- **Portability**—HDFS is portable across operating systems.
- **Write once**—By assuming a file will remain unchanged after it is written, HDFS simplifies replication and speeds up data throughput.
- **Locality of computation**—Due to data volume, it is often much faster to move the program near to the data, and HDFS has features to facilitate this.

HDFS provides an interface similar to that of regular filesystems. Unlike a database, HDFS can only store and retrieve data, not index it. Simple random access to data is not possible. However, higher-level layers have been created to provide finer-grained functionality to Hadoop deployments, such as HBase.

HBase, the Hadoop Database

One approach to making HDFS more usable is HBase. Modeled after Google's BigTable database, HBase is a column-oriented database designed to store massive amounts of data. It belongs to the NoSQL universe of databases, and is similar to Cassandra and Hypertable.



HBase uses HDFS as a storage system, and thus is capable of storing a large volume of data through fault-tolerant, distributed nodes. Like similar column-store databases, HBase provides REST and Thrift based API access.

Because it creates indexes, HBase offers fast, random access to its contents, though with simple queries. For complex operations, HBase acts as both a *source* and a *sink* (destination for computed data) for Hadoop MapReduce. HBase thus allows systems to interface with Hadoop as a database, rather than the lower level of HDFS.

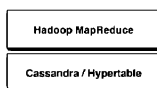
Hive

Data warehousing, or storing data in such a way as to make reporting and analysis easier, is an important application area for SMAQ systems. Developed originally at Facebook, Hive is a data warehouse framework built on top of Hadoop. Similar to HBase, Hive provides a table-based abstraction over HDFS and makes it easy to load structured data. In contrast to HBase, Hive can only run MapReduce jobs and is suited for batch data analysis. Hive provides a SQL-like query language to execute MapReduce jobs, described in the Query section below.

Cassandra and Hypertable

Cassandra and Hypertable are both scalable column-store databases that follow the pattern of BigTable, similar to HBase.

An Apache project, Cassandra originated at Facebook and is now in production in many large-scale websites, including Twitter, Facebook, Reddit and Digg. Hypertable was created at Zvents and spun out as an open source project.



Both databases offer interfaces to the Hadoop API that allow them to act as a source and a sink for MapReduce. At a higher level, Cassandra offers integration with the Pig query language (see the Query section below), and Hypertable has been integrated with Hive.

NoSQL database implementations of MapReduce

The storage solutions examined so far have all depended on Hadoop for MapReduce. Other NoSQL databases have built-in MapReduce features that allow computation to be parallelized over their data stores. In contrast with the multi-component SMAQ architectures of Hadoop-based systems, they offer a self-contained system comprising storage, MapReduce and query all in one.

Whereas Hadoop-based systems are most often used for batch-oriented analytical purposes, the usual function of NoSQL stores is to back live applications. The MapReduce functionality in these databases tends to be a secondary feature, augmenting other primary query mechanisms. Riak, for example, has a default timeout of 60 seconds on a MapReduce job, in contrast to the expectation of Hadoop that such a process may run for minutes or hours.

These prominent NoSQL databases contain MapReduce functionality:

- CouchDB is a distributed database, offering semi-structured document-based storage. Its key features include strong replication support and the ability to make distributed updates. Queries in CouchDB are implemented using JavaScript to define the map and reduce phases of a MapReduce process.
- MongoDB is very similar to CouchDB in nature, but with a stronger emphasis on performance, and less suitability for distributed updates, replication, and versioning. MongoDB MapReduce operations are specified using JavaScript.
- Riak is another database similar to CouchDB and MongoDB, but places its emphasis on high availability. MapReduce operations in Riak may be specified with JavaScript or Erlang.

Integration with SQL databases

In many applications, the primary source of data is in a relational database using platforms such as MySQL or Oracle. MapReduce is typically used with this data in two ways:

- Using relational data as a source (for example, a list of your friends in a social network).
- Re-injecting the results of a MapReduce operation into the database (for example, a list of product recommendations based on friends' interests).

It is therefore important to understand how MapReduce can interface with relational database systems. At the most basic level, delimited text files serve as an import and export format between relational databases and Hadoop systems, using a combination of SQL export commands and HDFS operations. More sophisticated tools do, however, exist.

The Sqoop tool is designed to import data from relational databases into Hadoop. It was developed by Cloudera, an enterprise-focused distributor of Hadoop platforms. Sqoop is database-agnostic, as it uses the Java JDBC database API. Tables can be imported either wholesale, or using queries to restrict the data import.

Sqoop also offers the ability to re-inject the results of MapReduce from HDFS back into a relational database. As HDFS is a filesystem, Sqoop expects delimited text files and transforms them into the SQL commands required to insert data into the database.

For Hadoop systems that utilize the Cascading API (see the Query section below) the `cascading.jdbc` and `cascading-dbmigrate` tools offer similar source and sink functionality.

Integration with streaming data sources

In addition to relational data sources, streaming data sources, such as web server log files or sensor output, constitute the most common source of input to big data systems. The Cloudera Flume project aims at providing convenient integration between Hadoop and streaming data sources. Flume aggregates data from both network and file sources, spread over a cluster of machines, and continuously pipes these into HDFS. The Scribe server, developed at Facebook, also offers similar functionality.

Commercial SMAQ solutions

Several massively parallel processing (MPP) database products have MapReduce functionality built in. MPP databases have a distributed architecture with independent nodes that run in parallel. Their primary application is in data warehousing and analytics, and they are commonly accessed using SQL.

- The Greenplum database is based on the open source PostgreSQL DBMS, and runs on clusters of distributed hardware. The addition of MapReduce to the regular SQL interface enables fast, large-scale analytics over Greenplum databases, reducing query times by several orders of magnitude. Greenplum MapReduce permits the mixing of external data sources with the database storage. MapReduce operations can be expressed as functions in Perl or Python.
- Aster Data's nCluster data warehouse system also offers MapReduce functionality. MapReduce operations are invoked using Aster Data's SQL-MapReduce technology. SQL-MapReduce enables the intermingling of SQL queries with MapReduce jobs defined using code, which may be written in languages including C#, C++, Java, R or Python.

Other data warehousing solutions have opted to provide connectors with Hadoop, rather than integrating their own MapReduce functionality.

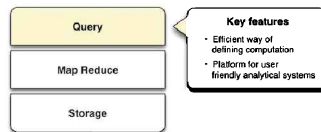
- Vertica, famously used by Farmville creator Zynga, is an MPP column-oriented database that offers a connector for Hadoop.
- Netezza is an established manufacturer of hardware data warehousing and analytical appliances. Recently acquired by IBM, Netezza is working with Hadoop distributor Cloudera to enhance the interoperation between their appliances and Hadoop. While it solves similar problems, Netezza falls outside of our SMAQ definition, lacking both the open source and commodity hardware aspects.

Although creating a Hadoop-based system can be done entirely with open source, it requires some effort to integrate such a system. Cloudera aims to make Hadoop enterprise-ready, and has created a unified Hadoop

distribution in its Cloudera Distribution for Hadoop (CDH). CDH for Hadoop parallels the work of Red Hat or Ubuntu in creating Linux distributions. CDH comes in both a free edition and an Enterprise edition with additional proprietary components and support. CDH is an integrated and polished SMAQ environment, complete with user interfaces for operation and query. Cloudera's work has resulted in some significant contributions to the Hadoop open source ecosystem.

Query

Specifying MapReduce jobs in terms of defining distinct map and reduce functions in a programming language is unintuitive and inconvenient, as is evident from the Java code listings shown above. To mitigate this, SMAQ systems incorporate a higher-level query layer to simplify both the specification of the MapReduce operations and the retrieval of the result.



Many organizations using Hadoop will have already written in-house layers on top of the MapReduce API to make its operation more convenient. Several of these have emerged either as open source projects or commercial products.

Query layers typically offer features that handle not only the specification of the computation, but the loading and saving of data and the orchestration of the processing on the MapReduce cluster. Search technology is often used to implement the final step in presenting the computed result back to the user.

Pig

Developed by Yahoo and now part of the Hadoop project, Pig provides a new high-level language, Pig Latin, for describing and running Hadoop MapReduce jobs. It is intended to make Hadoop accessible for developers familiar with data manipulation using SQL, and provides an

interactive interface as well as a Java API. Pig integration is available for the Cassandra and HBase databases.

Below is shown the word-count example in Pig, including both the data loading and storing phases (the notation $\$0$ refers to the first field in a record).

```
input = LOAD 'input/sentences.txt' USING TextLoader();
words = FOREACH input GENERATE
FLATTEN(TOKENIZE($0));grouped = GROUP words BY
$0;counts = FOREACH grouped GENERATE group,
COUNT(words);
ordered = ORDER counts BY $0;
STORE ordered INTO 'output/wordCount' USING
PigStorage();
```

While Pig is very expressive, it is possible for developers to write custom steps in User Defined Functions (UDFs), in the same way that many SQL databases support the addition of custom functions. These UDFs are written in Java against the Pig API.

Though much simpler to understand and use than the MapReduce API, Pig suffers from the drawback of being yet another language to learn. It is SQL-like in some ways, but it is sufficiently different from SQL that it is difficult for users familiar with SQL to reuse their knowledge.

Hive

As introduced above, Hive is an open source data warehousing solution built on top of Hadoop. Created by Facebook, it offers a query language very similar to SQL, as well as a web interface that offers simple query-building functionality. As such, it is suited for non-developer users, who may have some familiarity with SQL.

Hive's particular strength is in offering ad-hoc querying of data, in contrast to the compilation requirement of Pig and Cascading. Hive is a natural starting point for more full-featured business intelligence systems, which offer a user-friendly interface for non-technical users.

The Cloudera Distribution for Hadoop integrates Hive, and provides a higher-level user interface through the HUE project, enabling users to submit queries and monitor the execution of Hadoop jobs.

Cascading, the API Approach

The Cascading project provides a wrapper around Hadoop's MapReduce API to make it more convenient to use from Java applications. It is an intentionally thin layer that makes the integration of MapReduce into a larger system more convenient. Cascading's features include:

- A data processing API that aids the simple definition of MapReduce jobs.
- An API that controls the execution of MapReduce jobs on a Hadoop cluster.
- Access via JVM-based scripting languages such as Jython, Groovy, or JRuby.
- Integration with data sources other than HDFS, including Amazon S3 and web servers.
- Validation mechanisms to enable the testing of MapReduce processes.

Cascading's key feature is that it lets developers assemble MapReduce operations as a flow, joining together a selection of "pipes". It is well suited for integrating Hadoop into a larger system within an organization.

While Cascading itself doesn't provide a higher-level query language, a derivative open source project called Cascalog does just that. Using the Clojure JVM language, Cascalog implements a query language similar to that of Datalog. Though powerful and expressive, Cascalog is likely to remain a niche query language, as it offers neither the ready familiarity of Hive's SQL-like approach nor Pig's procedural expression. The listing below shows the word-count example in Cascalog: it is significantly terser, if less transparent.

```
(defmapcatop split [sentence]
  (seq (.split sentence "\\s+")))

(?<- (stdout) [?word ?count]
  (sentence ?s) (split ?s :> ?word)
  (c/count ?count))
```

Search with Solr

An important component of large-scale data deployments is retrieving and summarizing data. The addition of database layers such as HBase provides easier access to data, but does not provide sophisticated search capabilities.

To solve the search problem, the open source search and indexing platform Solr is often used alongside NoSQL database systems. Solr uses Lucene search technology to provide a self-contained search server product.

For example, consider a social network database where MapReduce is used to compute the influencing power of each person, according to some suitable metric. This ranking would then be reinjected to the database. Using Solr indexing allows operations on the social network, such as finding the most influential people whose interest profiles mention mobile phones, for instance.

Originally developed at CNET and now an Apache project, Solr has evolved from being just a text search engine to supporting faceted navigation and results clustering. Additionally, Solr can manage large data volumes over distributed servers. This makes it an ideal solution for result retrieval over big data sets, and a useful component for constructing business intelligence dashboards.

Conclusion

MapReduce, and Hadoop in particular, offers a powerful means of distributing computation among commodity servers. Combined with distributed storage and increasingly user-friendly query mechanisms, the

resulting SMAQ architecture brings big data processing within reach for even small- and solo-development teams.

It is now economic to conduct extensive investigation into data, or create data products that rely on complex computations. The resulting explosion in capability has forever altered the landscape of analytics and data warehousing systems, lowering the bar to entry and fostering a new generation of products, services and organizational attitudes—a trend explored more broadly in Mike Loukides' "What is Data Science?" report.

The emergence of Linux gave power to the innovative developer with merely a small Linux server at their desk: SMAQ has the same potential to streamline data centers, foster innovation at the edges of an organization, and enable new startups to cheaply create data-driven businesses.

Scraping, cleaning, and selling big data

Infochimps execs discuss the challenges of data scraping.

by Audrey Watters

In 2008, the Austin-based data startup Infochimps released a scrape of Twitter data that was later taken down at the request of the microblogging site because of user privacy concerns. Infochimps has since struck a deal with Twitter to make some datasets available on the site, and the Infochimps marketplace now contains more than 10,000 datasets from a variety of sources. Not all these datasets have been obtained via scraping, but nevertheless, the company's process of scraping, cleaning, and selling big data is an interesting topic to explore, both technically and legally.

With that in mind, Infochimps CEO Nick Ducoff, CTO Flip Kromer, and business development manager Dick Hall explain the business of data scraping in the following interview.

What are the legal implications of data scraping?

Dick Hall: There are three main areas you need to consider: copyright, terms of service, and “trespass to chattels.”

United States copyright law protects against unauthorized copying of “original works of authorship.” Facts and ideas are not copyrightable. However, expressions or arrangements of facts may be copyrightable. For example, a recipe for dinner is not copyrightable, but a recipe book with a series of recipes selected based on a unifying theme would be copyrightable. This example illustrates the “originality” requirement for copyright.

Let's apply this to a concrete web-scraping example. The New York Times publishes a blog post that includes the results of an election poll arranged in descending order by percentage. The New York Times can claim a copyright on the blog post, but not the table of poll results. A web scraper is free to copy the data contained in the table without fear of

copyright infringement. However, in order to make a copy of the blog post wholesale, the web scraper would have to rely on a defense to infringement, such as fair use. The result is that it is difficult to maintain a copyright over data, because only a specific arrangement or selection of the data will be protected.

Most websites include a page outlining their terms of service (ToS), which defines the acceptable use of the website. For example, YouTube forbids a user from posting copyrighted materials if the user does not own the copyright. Terms of service are based in contract law, but their enforceability is a gray area in US law. A web scraper violating the letter of a site's ToS may argue that they never explicitly saw or agreed to the terms of service.

Assuming ToS are enforceable, they are a risky issue for web scrapers. First, every site on the Internet will have a different ToS — Twitter, Facebook, and The New York Times may all have drastically different ideas of what is acceptable use. Second, a site may unilaterally change the ToS without notice and maintain that continued use represents acceptance of the new ToS by a web scraper or user. For example, Twitter recently changed its ToS to make it significantly more difficult for outside organizations to store or export tweets for any reason.

There's also the issue of volume. High-volume web scraping could cause significant monetary damages to the sites being scraped. For example, if a web scraper checks a site for changes several thousand times per second, it is functionally equivalent to a denial of service attack. In this case, the web scraper may be liable for damages under a theory of "trespass to chattels," because the site owner has a property interest in his or her web servers. A good-natured web scraper should be able to avoid this issue by picking a reasonable frequency for scraping.

What are some of the challenges of acquiring data through scraping?

Flip Kromer: There are several problems with the scale and the metadata, as well as historical complications.

- Scale — It's obvious that terabytes of data will cause problems, but so (on most filesystems) will having tens of millions of files in the same directory tree.
- Metadata — It's a chicken-and-egg problem. Since few programs can draw on rich metadata, it's not much use annotating it. But since so few datasets are annotated, it's not worth writing support into your applications. We have an internal data-description language that we plan to open source as it matures.
- Historical complications — Statisticians like SPSS files. Semantic web advocates like RDF/XML. Wall Street quants like Mathematica exports. There is no One True Format. Lifting each out of its source domain is time consuming.

But the biggest non-obvious problem we see is source domain complexity. This is what we call the “uber” problem. A developer wants the answer to a reasonable question, such as “What was the air temperature in Austin at noon on August 6, 1998?” The obvious answer — “damn hot” — isn't acceptable. Neither is:

Well, it's complicated. See, there are multiple weather stations, all reporting temperatures — each with its own error estimate — at different times. So you simply have to take the spatial- and time-average of their reported values across the region. And by the way, did you mean Austin's city boundary, or its metropolitan area, or its downtown region?

There are more than a dozen incompatible yet fundamentally correct ways to measure time: Earth-centered? Leap seconds? Calendrical? Does the length of a day change as the earth's rotational speed does?

Data at “everything” scale is sourced by domain experts, who necessarily live at the “it's complicated” level. To make it useful to the rest of the world requires domain knowledge, and often a transformation that is simply nonsensical within the source domain.

How will data marketplaces change the work and direction of data startups?

Nick Ducoff: I vividly remember being taught about comparative advantage. This might age me a bit, but the lesson was: Michael Jordan

doesn't mow his own lawn. Why? Because he should spend his time practicing basketball since that's what he's best at and makes a lot of money doing. The same analogy applies to software developers. If you are best at the presentation layer, you don't want to spend your time futzing around with databases

Infochimps allows these developers to spend their time doing what they do best — building apps — while we spend ours doing what we do best — making data easy to find and use. What we're seeing is startups focusing on pieces of the stack. Over time the big cloud providers will buy these companies to integrate into their stacks.

Companies like Heroku (acquired by Salesforce) and CloudKick (acquired by Rackspace) have paved the way for this. Tools like ScraperWiki and Junar will allow anybody to pull down tables off the web, and companies like Mashery, Apigee and 3scale will continue to make APIs more prevalent. We'll help make these tables and APIs findable and usable. Developers will be able to go from idea to app in hours, not days or weeks.

This interview was edited and condensed.

Data hand tools

A data task illustrates the importance of simple and flexible tools.

by Mike Loukides

The flowering of data science has both driven, and been driven by, an explosion of powerful tools. R provides a great platform for doing statistical analysis, Hadoop provides a framework for orchestrating large clusters to solve problems in parallel, and many NoSQL databases exist for storing huge amounts of unstructured data. The heavy machinery for serious number crunching includes perennials such as Mathematica, Matlab, and Octave, most of which have been extended for use with large clusters and other big iron.

But these tools haven't negated the value of much simpler tools; in fact, they're an essential part of a data scientist's toolkit. Hilary Mason and Chris Wiggins wrote that "Sed, awk, grep are enough for most small tasks," and there's a layer of tools below *sed*, *awk*, and *grep* that are equally useful. Hilary has pointed out the value of exploring data sets with simple tools before proceeding to a more in-depth analysis. The advent of cloud computing, Amazon's EC2 in particular, also places a premium on fluency with simple command-line tools. In conversation, Mike Driscoll of Metamarkets pointed out the value of basic tools like *grep* to filter your data before processing it or moving it somewhere else. Tools like *grep* were designed to do one thing and do it well. Because they're so simple, they're also extremely flexible, and can easily be used to build up powerful processing pipelines using nothing but the command line. So while we have an extraordinary wealth of power tools at our disposal, we'll be the poorer if we forget the basics.

With that in mind, here's a very simple, and not contrived, task that I needed to accomplish. I'm a ham radio operator. I spent time recently in a contest that involved making contacts with lots of stations all over the world, but particularly in Russia. Russian stations all sent their two-letter oblast abbreviation (equivalent to a US state). I needed to figure out how many oblasts I contacted, along with counting oblasts on particular ham

bands. Yes, I have software to do that; and no, it wasn't working (bad data file, since fixed). So let's look at how to do this with the simplest of tools.

(Note: Some of the spacing in the associated data was edited to fit on the page. If you copy and paste the data, a few commands that rely on counting spaces won't work.)

Log entries look like this:

```
QSO: 14000 CW 2011-03-19 1229 W1JQ      599 0001
UV5U      599 0041
QSO: 14000 CW 2011-03-19 1232 W1JQ      599 0002
SO2O      599 0043
QSO: 21000 CW 2011-03-19 1235 W1JQ      599 0003
RG3K      599 VR
QSO: 21000 CW 2011-03-19 1235 W1JQ      599 0004
UD3D      599 MO
...
```

Most of the fields are arcane stuff that we won't need for these exercises. The Russian entries have a two-letter oblast abbreviation at the end; rows that end with a number are contacts with stations outside of Russia. We'll also use the second field, which identifies a ham radio band (21000 KHz, 14000 KHz, 7000 KHz, 3500 KHz, etc.) So first, let's strip everything but the Russians with *grep* and a regular expression:

```
$ grep '599 [A-Z][A-Z]' ridx-log.txt | head -2
QSO: 21000 CW 2011-03-19 1235 W1JQ      599 0003
RG3K      599 VR
QSO: 21000 CW 2011-03-19 1235 W1JQ      599 0004
UD3D      599 MO
```

grep may be the most useful tool in the Unix toolchest. Here, I'm just searching for lines that have 599 (which occurs everywhere) followed by a space, followed by two uppercase letters. To deal with mixed case (not necessary here), use *grep -i*. You can use character classes like `:upper:` rather than specifying the range A-Z, but why bother? Regular expressions can become very complex, but simple will often do the job, and be less error-prone.

If you're familiar with *grep*, you may be asking why I didn't use `$` to match the end of line, and forget about the 599 noise. Good question. There is some whitespace at the end of the line; we'd have to match that, too. Because this file was created on a Windows machine, instead of just a newline at the end of each line, it has a return and a newline. The `$` that *grep* uses to match the end-of-line only matches a Unix newline. So I did the easiest thing that would work reliably.

The simple *head* utility is a jewel. If you leave *head* off of the previous command, you'll get a long listing scrolling down your screen. That's rarely useful, especially when you're building a chain of commands. *head* gives you the first few lines of output: 10 lines by default, but you can specify the number of lines you want. `-2` says "just two lines," which is enough for us to see that this script is doing what we want.

Next, we need to cut out the junk we don't want. The easy way to do this is to use *colrm* (remove columns). That takes two arguments: the first and last column to remove. Column numbering starts with one, so in this case we can use *colrm 1 72*.

```
$ grep '599 [A-Z][A-Z]' rudx-log.txt | colrm 1 72 |  
head -2  
VR  
MO  
...
```

How did I know we wanted column 72? Just a little experimentation; command lines are cheap, especially with command history editing. I should actually use 73, but that additional space won't hurt, nor will the additional whitespace at the end of each line. Yes, there are better ways to select columns; we'll see them shortly. Next, we need to sort and find the unique abbreviations. I'm going to use two commands here: *sort* (which does what you'd expect), and *uniq* (to remove duplicates).

```
$ grep '599 [A-Z][A-Z]' rudx-log.txt | colrm 1 72 |  
sort |\n  uniq | head -2  
AD  
AL
```

Sort has a *-u* option that suppresses duplicates, but for some reason I prefer to keep *sort* and *uniq* separate. *sort* can also be made case-insensitive (*-f*), can select particular fields (meaning we could eliminate the *colrm* command, too), can do numeric sorts in addition to lexical sorts, and lots of other things. Personally, I prefer building up long Unix pipes one command at a time to hunting for the right options.

Finally, I said I wanted to count the number of oblasts. One of the most useful Unix utilities is a little program called *wc*: “word count.” That’s what it does. Its output is three numbers: the number of lines, the number of words, and the number of characters it has seen. For many small data projects, that’s really all you need.

```
$ grep '599 [A-Z][A-Z]' rudx-log.txt | colrm 1 72 |
sort | uniq | wc
      38          38         342
```

So, 38 unique oblasts. You can say *wc -l* if you only want to count the lines; sometimes that’s useful. Notice that we no longer need to end the pipeline with *head*; we want *wc* to see all the data.

But I said I also wanted to know the number of oblasts on each ham band. That’s the first number (like 21000) in each log entry. So we’re throwing out too much data. We could fix that by adjusting *colrm*, but I promised a better way to pull out individual columns of data. We’ll use *awk* in a very simple way:

```
$ grep '599 [A-Z][A-Z]' rudx-log.txt | awk '{print $2
" " $11}' | \
sort | uniq
14000 AD
14000 AL
14000 AN
...
```

awk is a very powerful tool; it’s a complete programming language that can do almost any kind of text manipulation. We could do everything we’ve seen so far as an *awk* program. But rather than use it as a power tool, I’m just using it to pull out the second and eleventh fields from my

input. The single quotes are needed around the *awk* program, to prevent the Unix shell from getting confused. Within *awk*'s print command, we need to explicitly include the space, otherwise it will run the fields together.

The *cut* utility is another alternative to *colrm* and *awk*. It's designed for removing portions of a file. *cut* isn't a full programming language, but it can make more complex transformations than simply deleting a range of columns. However, although it's a simple tool at heart, it can get tricky; I usually find that, when *colrm* runs out of steam, it's best jumping all the way to *awk*.

We're still a little short of our goal: how do we count the number of oblasts on each band? At this point, I use a really cheesy solution: another *grep*, followed by *wc*:

```
$ grep '599 [A-Z][A-Z]' rudx-log.txt | awk '{print $2
" " $11}' | \
    sort | uniq | grep 21000 | wc
    20      40      180
$ grep '599 [A-Z][A-Z]' rudx-log.txt | awk '{print $2
" " $11}' | \
    sort | uniq | grep 14000 | wc
    26      52      234
...
```

OK, 20 oblasts on the 21 MHz band, 26 on the 14 MHz band. And at this point, there are two questions you really should be asking. First, why not put *grep* 21000 first, and save the *awk* invocation? That's just how the script developed. You could put the *grep* first, though you'd still need to strip extra gunk from the file. Second: What if there are gigabytes of data? You have to run this command for each band, and for some other project, you might need to run it dozens or hundreds of times. That's a valid objection. To solve this problem, you need a more complex *awk* script (which has associative arrays in which you can save data), or you need a programming language such as perl, python, or ruby. At the same time, we've gotten fairly far with our data exploration, using only the simplest of tools.

Now let's up the ante. Let's say that there are a number of directories with lots of files in them, including these `rudx-log.txt` files. Let's say that these directories are organized by year (2001, 2002, etc.). And let's say we want to count oblasts across all the years for which we have records. How do we do that?

Here's where we need *find*. My first approach is to take the filename (`rudx-log.txt`) out of the *grep* command, and replace it with a *find* command that looks for every file named `rudx-log.txt` in subdirectories of the current directory:

```
$ grep '599 [A-Z][A-Z]' `find . -name rudx-log.txt`
-print ` | \
    awk '{print $2 " " $11}' | sort | uniq | grep 14000
| wc
      48          96          432
```

OK, so 48 directories on the 14 MHz band, lifetime. I thought I had done better than that. What's happening, though? That *find* command is simply saying "look at the current directory and its subdirectories, find files with the given name, and print the output." The backquotes tell the Unix shell to use the output of *find* as arguments to *grep*. So we're just giving *grep* a long list of files, instead of just one. Note the `-print` option: if it's not there, *find* happily does nothing.

We're almost done, but there are a couple of bits of hair you should worry about. First, if you invoke *grep* with more than one file on the command line, each line of output begins with the name of the file in which it found a match:

```
...
./2008/rudx-log.txt:QSO: 14000 CW 2008-03-15 1526
W1JQ      599 0054 \\
UA6YW      599 AD
./2009/rudx-log.txt:QSO: 14000 CW 2009-03-21 1225
W1JQ      599 0015 \\
RG3K      599 VR
...
```

We're lucky. *grep* just sticks the filename at the beginning of the line without adding spaces, and we're using *awk* to print selected whitespace-separated fields. So the number of any field didn't change. If we were using *colrm*, we'd have to fiddle with things to find the right columns. If the filenames had different lengths (reasonably likely, though not possible here), we couldn't use *colrm* at all. Fortunately, you can suppress the filename by using *grep -h*.

The second piece of hair is less common, but potentially more troublesome. If you look at the last command, what we're doing is giving the *find* command a really long list of filenames. How long is long? Can that list get too long? The answers are "we don't know," and "maybe." In the nasty old days, things broke when the command line got longer than a few thousand characters. These days, who knows what's too long ... But we're doing "big data," so it's easy to imagine the *find* command expanding to hundreds of thousands, even millions of characters. More than that, our single Unix pipeline doesn't parallelize very well; and if we really have big data, we want to parallelize it.

The answer to this problem is another old Unix utility, *xargs*. *Xargs* dates back to the time when it was fairly easy to come up with file lists that were too long. Its job is to break up command line arguments into groups and spawn as many separate commands as needed, running in parallel if possible (*-P*). We'd use it like this:

```
$ find . -name ridx-log.txt -print | xargs grep '599
[A-Z][A-Z]' | \
  awk '{print $2 " " $11}' | grep 14000 | sort | uniq
| wc
    48      96     432
```

This command is actually a nice little map-reduce implementation: the *xargs* command maps *grep* all the cores on your machine, and the output is reduced (combined) by the *awk/sort/uniq* chain. *xargs* has lots of command line options, so if you want to be confused, read the man page.

Another approach is to use *find*'s *-exec* option to invoke arbitrary commands. It's somewhat more flexible than *xargs*, though in my

opinion, *find -exec* has the sort of overly flexible but confusing syntax that's surprisingly likely to lead to disaster. (It's worth noting that the examples for *-exec* almost always involve automating bulk file deletion. Excuse me, but that's a recipe for heartache. Take this from the guy who once deleted the business plan, then found that the backups hadn't been done for about 6 months.) There's an excellent tutorial for both *xargs* and *find -exec* at Softpanorama. I particularly like this tutorial because it emphasizes testing to make sure that your command won't run amok and do bad things (like deleting the business plan).

That's not all. Back in the dark ages, I wrote a shell script that did a recursive *grep* through all the subdirectories of the current directory. That's a good shell programming exercise which I'll leave to the reader. More to the point, I've noticed that there's now a *-R* option to *grep* that makes it recursive. Clever little buggers ...

Before closing, I'd like to touch on a couple of tools that are a bit more exotic, but which should be in your arsenal in case things go wrong. *od -c* gives a raw dump of every character in your file. (*-c* says to dump characters, rather than octal or hexadecimal). It's useful if you think your data is corrupted (it happens), or if it has something in it that you didn't expect (it happens a LOT). *od* will show you what's happening; once you know what the problem is, you can fix it. To fix it, you may want to use *sed*. *sed* is a cranky old thing: more than a hand tool, but not quite a power tool; sort of an antique treadle-operated drill press. It's great for editing files on the fly, and doing batch edits. For example, you might use it if NUL characters were scattered through the data.

Finally, a tool I just learned about (thanks, @dataspora): the pipe viewer, *pv*. It isn't a standard Unix utility. It comes with some versions of Linux, but the chances are that you'll have to install it yourself. If you're a Mac user, it's in macports. *pv* tells you what's happening inside the pipes as the command progresses. Just insert it into a pipe like this:

```
$ find . -name ridx-log.txt -print | xargs grep '599
[A-Z][A-Z]' | \
  awk '{print $2 " " $11}' | pv | grep 14000 | sort |
uniq | wc
```

The pipeline runs normally, but you'll get some additional output that shows the command's progress. If something's getting malfunctioning or performing too slowly, you'll find out. *pv* is particularly good when you have huge amounts of data, and you can't tell whether something has ground to a halt, or you just need to go out for coffee while the command runs to completion.

Whenever you need to work with data, don't overlook the Unix "hand tools." Sure, everything I've done here could be done with Excel or some other fancy tool like R or Mathematica. Those tools are all great, but if your data is living in the cloud, using these tools is possible, but painful. Yes, we have remote desktops, but remote desktops across the Internet, even with modern high-speed networking, are far from comfortable. Your problem may be too large to use the hand tools for final analysis, but they're great for initial explorations. Once you get used to working on the Unix command line, you'll find that it's often faster than the alternatives. And the more you use these tools, the more fluent you'll become.

Oh yeah, that broken data file that would have made this exercise superfluous? Someone emailed it to me after I wrote these scripts. The scripting took less than 10 minutes, start to finish. And, frankly, it was more fun.

Hadoop: What it is, how it works, and what it can do

Cloudera CEO Mike Olson on Hadoop's architecture and its data applications.

by James Turner

Hadoop gets a lot of buzz these days in database and content management circles, but many people in the industry still don't really know what it is and or how it can be best applied.

Cloudera CEO and Strata speaker Mike Olson, whose company offers an enterprise distribution of Hadoop and contributes to the project, discusses Hadoop's background and its applications in the following interview.

Where did Hadoop come from?

Mike Olson: The underlying technology was invented by Google back in their earlier days so they could usefully index all the rich textural and structural information they were collecting, and then present meaningful and actionable results to users. There was nothing on the market that would let them do that, so they built their own platform. Google's innovations were incorporated into Nutch, an open source project, and Hadoop was later spun-off from that. Yahoo has played a key role developing Hadoop for enterprise applications.

What problems can Hadoop solve?

Mike Olson: The Hadoop platform was designed to solve problems where you have a lot of data — perhaps a mixture of complex and structured data — and it doesn't fit nicely into tables. It's for situations where you want to run analytics that are deep and computationally extensive, like clustering and targeting. That's exactly what Google was doing when it was indexing the web and examining user behavior to improve performance algorithms.

Hadoop applies to a bunch of markets. In finance, if you want to do accurate portfolio evaluation and risk analysis, you can build sophisticated models that are hard to jam into a database engine. But Hadoop can handle it. In online retail, if you want to deliver better search answers to your customers so they're more likely to buy the thing you show them, that sort of problem is well addressed by the platform Google built. Those are just a few examples.

How is Hadoop architected?

Mike Olson: Hadoop is designed to run on a large number of machines that don't share any memory or disks. That means you can buy a whole bunch of commodity servers, slap them in a rack, and run the Hadoop software on each one. When you want to load all of your organization's data into Hadoop, what the software does is bust that data into pieces that it then spreads across your different servers. There's no one place where you go to talk to all of your data; Hadoop keeps track of where the data resides. And because there are multiple copy stores, data stored on a server that goes offline or dies can be automatically replicated from a known good copy.

In a centralized database system, you've got one big disk connected to four or eight or 16 big processors. But that is as much horsepower as you can bring to bear. In a Hadoop cluster, every one of those servers has two or four or eight CPUs. You can run your indexing job by sending your code to each of the dozens of servers in your cluster, and each server operates on its own little piece of the data. Results are then delivered back to you in a unified whole. That's MapReduce: you map the operation out to all of those servers and then you reduce the results back into a single result set.

Architecturally, the reason you're able to deal with lots of data is because Hadoop spreads it out. And the reason you're able to ask complicated computational questions is because you've got all of these processors, working in parallel, harnessed together.

At this point, do companies need to develop their own Hadoop applications?

Mike Olson: It's fair to say that a current Hadoop adopter must be more sophisticated than a relational database adopter. There are not that many "shrink wrapped" applications today that you can get right out of the box and run on your Hadoop processor. It's similar to the early '80s when Ingres and IBM were selling their database engines and people often had to write applications locally to operate on the data.

That said, you can develop applications in a lot of different languages that run on the Hadoop framework. The developer tools and interfaces are pretty simple. Some of our partners — Informatica is a good example — have ported their tools so that they're able to talk to data stored in a Hadoop cluster using Hadoop APIs. There are specialist vendors that are up and coming, and there are also a couple of general process query tools: a version of SQL that lets you interact with data stored on a Hadoop cluster, and Pig, a language developed by Yahoo that allows for data flow and data transformation operations on a Hadoop cluster.

Hadoop's deployment is a bit tricky at this stage, but the vendors are moving quickly to create applications that solve these problems. I expect to see more of the shrink-wrapped apps appearing over the next couple of years.

Where do you stand in the SQL vs NoSQL debate?

Mike Olson: I'm a deep believer in relational databases and in SQL. I think the language is awesome and the products are incredible.

I hate the term "NoSQL." It was invented to create cachet around a bunch of different projects, each of which has different properties and behaves in different ways. The real question is, what problems are you solving? That's what matters to users.

Four free data tools for journalists (and snoops)

A look at free services that reveal traffic data, server details and popularity.

by Pete Warden

Note: The following is an excerpt from Pete Warden's free ebook "Where are the bodies buried on the web? Big data for journalists."

There's been a revolution in data over the last few years, driven by an astonishing drop in the price of gathering and analyzing massive amounts of information. It only cost me \$120 to gather, analyze and visualize 220 million public Facebook profiles, and you can use 80legs to download a million web pages for just \$2.20. Those are just two examples.

The technology is also getting easier to use. Companies like Extractiv and Needlebase are creating point-and-click tools for gathering data from almost any site on the web, and every other stage of the analysis process is getting radically simpler too.

What does this mean for journalists? You no longer have to be a technical specialist to find exciting, convincing and surprising data for your stories. For example, the following four services all easily reveal underlying data about web pages and domains.

WHOIS

Many of you will already be familiar with WHOIS, but it's so useful for research it's still worth pointing out. If you go to this site (or just type "whois www.example.com" in Terminal.app on a Mac) you can get the basic registration information for any website. In recent years, some owners have chosen "private" registration, which hides their details from view, but in many cases you'll see a name, address, email and phone number for the person who registered the site.

You can also enter numerical IP addresses here and get data on the organization or individual that owns that server. This is especially handy when you're trying to track down more information on an abusive or malicious user of a service, since most websites record an IP address for everyone who accesses them

Blekkio

The newest search engine in town, one of Blekkio's selling points is the richness of the data it offers. If you type in a domain name followed by /seo, you'll receive a page of statistics on that URL:

The screenshot shows the Blekkio search engine interface. At the top, the search bar contains 'petewarden.typepad.com /seo'. Below the search bar, there is a search button with a magnifying glass icon. The main content area displays a table of inbound links for the specified domain.

blekko .co.uk
petewarden.typepad.com /seo
examples: cure for headaches | global warming | internet

Search

Inbound links: 6,050 from 302 domains:

#	from host	host rank	links	last	actions
1	twitter.com	12,366.4	1		
2	www.guardian.co.uk	6,481.2	1		
3	www.fortles.com	3,699.8	1	416 ago	
4	www.newseintit.com	3,678.4	2		
5	code.google.com	3,451.1	1		
6	www.huffingtonpost.com	3,238.2	1		
7	news.cnet.com	3,185.8	2		
8	gizmodo.com	2,119.3	6	395 ago	

The first tab shows other sites that are linking to the current domain, in popularity order. This can be extremely useful when you're trying to understand what coverage a site is receiving, and if you want to understand why it's ranking highly in Google's search results, since they're based on those inbound links. Inclusion of this information would have been an interesting addition to the recent DecorMyEyes story, for example.

The other handy tab is "Crawl stats," especially the "Cohosted with" section:

The screenshot shows the 'Cohosted With' section of the Blekkio interface. It displays a list of domains that are cohosted with the current domain, along with their hostnames and whois information.

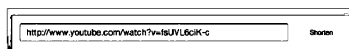
Cohosted With:	host	whois	view
	thelongtail.com	whois	
	oodinghorror.com	whois	
	longtail.com	whois	
	cityofsound.com	whois	
	hypebot.com	whois	
	therestisnoise.com	whois	
	steverberlinjohnson.com	whois	
	planeout.com	whois	
	riehworldview.com	whois	

This tells you which other websites are running from the same machine. It's common for scammers and spammers to astroturf their way toward

legitimacy by building multiple sites that review and link to each other. They look like independent domains, and may even have different registration details, but often they'll actually live on the same server because that's a lot cheaper. These statistics give you an insight into the hidden business structure of shady operators.

bit.ly

I always turn to bit.ly when I want to know how people are sharing a particular link. To use it, enter the URL you're interested in:



Then click on the 'Info Page+' link:

http://bit.ly/akOako Continue | Back Info Page+

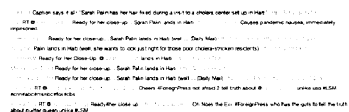
That takes you to the full statistics page (though you may need to choose "aggregate bit.ly link" first if you're signed in to the service).

31 Total Clicks
All clicks on the aggregate bit.ly link bit.ly/akOako

Long Links: <http://www.dailymail.co.uk/news/article-1337939/Sarah-Palin-0000.html>

Conversations: **Tweets 21**, **Shares 63**, **Likes 38**, **Comments 140**, **Stares 0**

This will give you an idea of how popular the page is, including activity on Facebook and Twitter. Below that you'll see public conversations about the link provided by backtype.com.



I find this combination of traffic data and conversations very helpful when I'm trying to understand why a site or page is popular, and who exactly its fans are. For example, it provided me with strong evidence that the prevailing narrative about grassroots sharing and Sarah Palin was wrong.

[Disclosure: O'Reilly AlphaTech Ventures is an investor in bit.ly.]

Compete

By surveying a cross-section of American consumers, Compete builds up detailed usage statistics for most websites, and they make some basic details freely available.

Choose the “Site Profile” tab and enter a domain:



You'll then see a graph of the site's traffic over the last year, together with figures for how many people visited, and how often.



Since they're based on surveys, Compete's numbers are only approximate. Nonetheless, I've found them reasonably accurate when I've been able to compare them against internal analytics.

Compete's stats are a good source when comparing two sites. While the absolute numbers may be off for both sites, Compete still offers a decent representation of the sites' relative difference in popularity.

One caveat: Compete only surveys U.S. consumers, so the data will be poor for predominantly international sites.

Additional data resources and tools are discussed in Pete's free ebook.

The quiet rise of machine learning

Alasdair Allan on how machine learning is taking over the mainstream.

by Jenn Webb

The concept of machine learning was brought to the forefront for the general masses when IBM's Watson computer appeared on Jeopardy and wiped the floor with humanity. For those same masses, machine learning quickly faded from view as Watson moved out of the spotlight ... or so they may think.

Machine learning is slowly and quietly becoming democratized. Goodreads, for instance, recently purchased Discovereads.com, presumably to make use of its machine learning algorithms to make book recommendations.

To find out more about what's happening in this rapidly advancing field, I turned to Alasdair Allan, an author and senior research fellow in Astronomy at the University of Exeter. In an email interview, he talked about how machine learning is being used behind the scenes in everyday applications. He also discussed his current eSTAR intelligent robotic telescope network project and how that machine learning-based system could be used in other applications.

In what ways is machine learning being used?

Alasdair Allan: Machine learning is quietly taking over in the mainstream. Orbitz, for instance, is using it behind the scenes to optimize caching of hotel prices, and Google is going to roll out smarter advertisements — much of the machine learning that consumers are seeing and using every day is invisible to them.

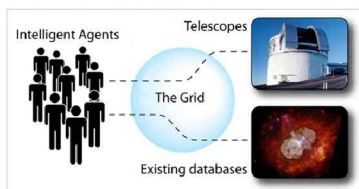
The interesting thing about machine learning right now is that research in the field is going on quietly as well because large corporations are tied up in non-disclosure agreements. While there is a large amount of

academic literature on the subject, it's actually hard to tell whether this open research is actually current.

Oddly, machine learning research mirrors the way cryptography research developed around the middle of the 20th century. Much of the cutting edge research was done in secret, and we're only finding out now, 40 or 50 years later, what GCHQ or the NSA was doing back then. I'm hopeful that it won't take quite that long for Amazon or Google to tell us what they're thinking about today.

How does your eSTAR intelligent robotic telescope network work?

Alasdair Allan: My work has focused on applying intelligent agent architectures and techniques to astronomy for telescope control and scheduling, and also for data mining. I'm currently leading the work at Exeter building a peer-to-peer distributed network of telescopes that, acting entirely autonomously, can reactively schedule observations of time-critical transient events in real-time. Notable successes include contributing to the detection of the most distant object yet discovered, a gamma-ray burster at a redshift of 8.2.



A diagram showing how the eSTAR network operates. The Intelligent Agents access telescopes and existing astronomical databases through the Grid. CREDIT: Joint Astronomy Centre. Eta Carinae image courtesy of N. Smith (U. Colorado), J. Morse (Arizona State U.), and NASA.

All the components of the system are thought of as agents — effectively “smart” pieces of software. Negotiation takes place between the agents in the system. each of the resources bids to carry out the work, with the science agent scheduling the work with the agent embedded at the resource that promises to return the best result.

This architectural distinction of viewing both sides of the negotiation as agents — and as equals — is crucial. Importantly, this preserves the autonomy of individual resources to implement observation scheduling at their facilities as they see fit, and it offers increased adaptability in the face of asynchronously arriving data.

The system is a meta-network that layers communication, negotiation, and real-time analysis software on top of existing telescopes, allowing scheduling and prioritization of observations to be done locally. It is flat, peer-to-peer, and owned and operated by disparate groups with their own goals and priorities. There is no central master-scheduler overseeing the network — optimization arises through emerging complexity and social convention.

How could the ideas behind eSTAR be applied elsewhere?

Alasdair Allan: Essentially what I've built is a geographically distributed sensor architecture. The actual architectures I've used to do this are entirely generic — fundamentally, it's just a peer-to-peer distributed system for optimizing scarce resources in real-time in the face of a constantly changing environment.

The architectures are therefore equally applicable to other systems. The most obvious use case is sensor motes. Cheap, possibly even disposable, single-use, mesh-networked sensor bundles could be distributed over a large geographic area to get situational awareness quickly and easily. Despite the underlying hardware differences, the same distributed machine learning-based architectures can be used.

At February's Strata conference, Alasdair Allan discussed the ambiguity surrounding a formal definition of machine learning:

<http://youtube.com>

This interview was edited and condensed.

Where the semantic web stumbled, linked data will succeed

Linked data allows for deep and serendipitous consumer experiences.

by Tyler Bell

In the same way that the Holy Roman Empire was neither holy nor Roman, Facebook's OpenGraph Protocol is neither open nor a protocol. It is, however, an extremely straightforward and applicable standard for document metadata. From a strictly semantic viewpoint, OpenGraph is considered hardly worthy of comment: it is a frankenstandard, a mishmash of microformats and loosely-typed entities, lobbed casually into the semantic web world with hardly a backward glance.

But this is not important. While OpenGraph avoids, or outright ignores, many of the problematic issues surrounding semantic annotation (see Alex Iskold's excellent commentary on OpenGraph here on Radar), criticism focusing only on its technical purity is missing half of the equation. Facebook gets it right where other initiatives have failed. While OpenGraph is incomplete and imperfect, it is immediately usable and sympathetic with extant approaches. Most importantly, OpenGraph is one component in a wider ecosystem. Its deployment benefits are apparent to the consumer and the developer: add the metatags, get the "likes," know your customers.

Such consumer causality is critical to the adoption of any semantic markup. We've seen it before with microformats, whose eventual popularity was driven by their ability to improve how a page is represented in search engine listings, and not by an abstract desire to structure the unstructured. Successful adoption will often entail sacrificing standardization and semantic purity for pragmatic ease-of-use; this is where the semantic web appears to have stumbled, and where linked data will most likely succeed.

Linked data intends to make the Web more interconnected and data-oriented. Beyond this outcome, the term is less rigidly defined. I would argue that linked data is more of an ethos than a standard, focused on providing context, assisting in disambiguation, and increasing serendipity within the user experience. This idea of linked data can be delivered by a number of existing components that work together on the data, platform, and application levels:

- **Entity provision:** Defining the who, what, where and when of the Internet, entities encapsulate meaning and provide context by type. In its most basic sense, an entity is one row in a list of things organized by type—such as people, places, or products—each with a unique identifier. Organizations that realize the benefits of linked data are releasing entities like never before, including the publication of 10,000 subject headings by the New York Times, admin regions and postcodes from the UK’s Ordnance Survey, placenames from Yahoo GeoPlanet, and the data infrastructures being created by Factual [disclosure: I’ve just signed on with Factual].
- **Entity annotation:** There are numerous formats for annotating entities when they exist in unstructured content, such as a web page or blog post. Facebook’s OpenGraph is a form of entity annotation, as are HTML5 microdata, RDFa, and microformats such as hCard. Microdata is the shiny, new player in the game, but see Evan Prodromou’s great post on RDFa v. microformats for a breakdown of these two more established approaches.
- **Endpoints and Introspection:** Entities contribute best to a linked data ecosystem when each is associated with a Uniform Resource Identifier (URI), an Internet-accessible, machine readable endpoint. These endpoints should provide *introspection*, the means to obtain the properties of that entity, including its relationship to others. For example, the Ordnance Survey URI for the “City of Southampton” is <http://data.ordnancesurvey.co.uk/id/7000000000037256>. Its properties can be retrieved in machine-readable format (RDF/XML, Turtle and JSON) by appending an “rdf,” “ttl,” or “json” extension to the above. To be properly open, URIs must be

accessible outside a formal API and authentication mechanism, exposed to semantically-aware web crawlers and search tools such as Yahoo BOSS. Under this definition, local business URLs, for example, can serve in-part as URIs—‘view source’ to see the semi-structured data in these listings from Yelp (using hcard and OpenGraph), and Foursquare (using microdata and OpenGraph).

- **Entity extraction:** Some linked data enthusiasts long for the day when all content is annotated so that it can be understood equally well by machines and humans. Until we get to that happy place, we will continue to rely on entity extraction technologies that parse unstructured content for recognizable entities, and make contextually intelligent identifications of their type and identifier. Named entity recognition (NER) is one approach that employs the above entity lists, which may also be combined with heuristic approaches designed to recognize entities that lie outside of a known entity list. Yahoo, Google and Microsoft are all hugely interested in this area, and we’ll see an increasing number of startups like Semantinet emerge with ever-improving precision and recall. If you want to see how entity extraction works first-hand, check out Reuters-owned Open Calais and experiment with their form-based tool.
- **Entity concordance and crosswalking:** The multitude of place namespaces illustrates how a single entity, such as a local business, will reside in multiple lists. Because the “unique” (U) in a URI is unique only to a given namespace, a world driven by linked data requires systems that explicitly match a single entity across namespaces. Examples of crosswalking services include: Placecast’s Match API, which returns the Placecast IDs of any place when supplied with an hcard equivalent; Yahoo’s Concordance, which returns the Where on Earth Identifier (WOEID) of a place using as input the place ID of one of fourteen external resources, including OpenStreetMap and Geonames; and the Guardian Content API, which allows users to search Guardian content using non-Guardian identifiers. These systems are the unsung heroes of the linked data world, facilitating interoperability by establishing links between

identical entities across namespaces. Huge, unrealized value exists within these applications, and we need more of them.

- **Relationships:** Entities are only part of the story. The real power of the semantic web is realized in knowing how entities of different types relate to each other: actors to movies, employees to companies, politicians to donors, restaurants to neighborhoods, or brands to stores. The power of all graphs—these networks of entities—is not in the entities themselves (the nodes), but how they relate together (the edges). However, I may be alone in believing that we need to nail the problem of multiple instances of the same entity, via concordance and crosswalking, before we can tap properly into the rich vein that entity relationships offer.

The approaches outlined above combine to help publishers and application developers provide intelligent, deep and serendipitous consumer experiences. Examples include the semantic handset from Aro Mobile, the BBC's World Cup experience, and aggregating references on your Facebook news feed.

Linked data will triumph in this space because efforts to date focus less on the *how* and more on the *why*. RDF, SPARQL, OWL, and triple stores are onerous. URIs, micro-formats, RDFa, and JSON, less so. Why invest in difficult technologies if consumer outcomes can be realized with extant tools and knowledge? We have the means to realize linked data now—the pieces of the puzzle are there and we (just) need to put them together.

Linked data is, at last, bringing the discussion around to the user. The consumer “end” trumps the semantic “means.”

Social data is an oracle waiting for a question

“Mining the Social Web” author Matthew Russell on the questions and answers social data can handle.

by Mac Slocum

We’re still in the stage where access to massive amounts of social data has novelty. That’s why companies are pumping out APIs and services are popping up to capture and sort all that information. But over time, as the novelty fades and the toolsets improve, we’ll move into a new phase that’s defined by the *application* of social data. Access will be implied. It’s what you do with the data that will matter.

Matthew Russell (@ptwobrussell), author of “Mining the Social Web” and a speaker at the upcoming Where 2.0 Conference, has already rounded that corner. In the following interview, Russell discusses the tools and the mindset that can unlock social data’s real utility.

How do you define the “social web”?

Matthew Russell: The “social web” is admittedly a notional entity with some blurry boundaries. There isn’t a Venn diagram that carves the “social web” out of the overall web fabric. The web is inherently a social fabric, and it’s getting more social all the time.

The distinction I make is that some parts of the fabric are much easier to access than others. Naturally, the platforms that expose their data with well-defined APIs will be the ones to receive the most attention and capture the mindshare when someone thinks of the “social web.”

In that regard, the social web is more of a heatmap where the hot areas are popular social networking hubs like Twitter, Facebook, and LinkedIn. Blogs, mailing lists, and even source code repositories such as Source Forge GitHub, however, are certainly part of the social web.

What sorts of questions can social data answer?

Matthew Russell: Here are some concrete examples of questions I asked — and answered — in “Mining the Social Web”:

- What’s your potential influence when you tweet?
- What does Justin Bieber have (or not have) in common with the Tea Party?
- Where does most of your professional network geographically reside, and how might this impact career decisions?
- How do you summarize the content of blog posts to quickly get the gist?
- Which of your friends on Twitter, Facebook, or elsewhere know one another, and how well?

It’s not hard at all to ask lots of valuable questions against social web data and answer them with high degrees of certainty. The most popular sources of social data are popular because they’re generally platforms that expose the data through well-crafted APIs. The effect is that it’s fairly easy to amass the data that you need to answer questions.

With the necessary data in hand to answer your questions, the selection of a programming language, toolkit, and/or framework that makes shaking out the answer is a critical step that shouldn’t be taken lightly. The more efficient it is to test your hypotheses, the more time you can spend *analyzing* your data. Spending sufficient time in analysis engenders the kind of creative freedom needed to produce truly interesting results. This why organizations like Infochimps and GNIP are filling a critical void.

What programming skills or development background do you need to effectively analyze social data?

Matthew Russell: A basic programming background definitely helps, because it allows you to automate so many of the mundane tasks that are involved in getting the data and munging it into a normalized form that’s easy to work with. That said, the lack of a programming background should be among the last things that stops you from diving head first into social data analysis. If you’re sufficiently motivated and analytical

enough to ask interesting questions, there's a very good chance you can pick up an easy language, like Python or Ruby, and learn enough to be dangerous over a weekend. The rest will take care of itself.

Why did you opt to use GitHub to share the example code from the book?

Matthew Russell: GitHub is a fantastic source code management tool, but the most interesting thing about it is that it's a *social* coding repository. What GitHub allows you to do is share code in such a way that people can clone your code repository. They can make improvements or fork the examples into an entirely new form, and then share those changes with the rest of the world in a very transparent way.

If you look at the project I started on GitHub, you can see exactly who did what with the code, whether I incorporated their changes back into my own repository, whether someone else has done something novel by using an example listing as a template, etc. You end up with a community of people that emerge around common causes, and amazing things start to happen as these people share and communicate about important problems and ways to solve them.

While I of course want people buy the book, all of the source code is out there for the taking. I hope people put it to good use.

The challenges of streaming real-time data

Jud Valeski on how Gnip handles the Twitter fire hose.

by Audrey Watters

Although Gnip handles real-time streaming of data from a variety of social media sites, it's best known as the official commercial provider of the Twitter activity stream.

Frankly, "stream" is a misnomer. "Fire hose," the colloquial variation, better represents the torrent of data Twitter produces. That hose pumps out around 155 million tweets per day, and it's all addressed at a sustained rate.

I recently spoke with Gnip CEO Jud Valeski (@jvaleski) about what it takes to manage Twitter's flood of data and how the Internet's architecture needs to adapt to real-time needs. Our interview follows.

The Internet wasn't really built to handle a river of big data. What are the architectural challenges of running real-time data through these pipes?

Jud Valeski: The most significant challenge is rusty infrastructure. Just as with many massive infrastructure projects that the world has seen, adopted, and exploited (aqueducts, highways, power/energy grids), the connective tissue of the network becomes excruciatingly dated. We're lucky to have gotten as far as we have on it. The capital build-outs on behalf of the telecommunications industry have yielded relatively low-bandwidth solutions laden with false advertising about true throughput. The upside is that highly transactional HTTP REST apps are relatively scalable in this environment and they "just work." It isn't until we get into heavy payload apps — video streaming, large-scale activity fire hoses like Twitter — that the deficiencies in today's network get put in the spotlight. That's when the pipes begin to burst.

We can redesign applications to create smaller activities/actions in order to reduce overall sizes. We can use tighter protocols/formats (Protocol Buffers for example), and compression to minimize sizes as well. However, with the ever-increasing usage of social networks generating more “activities,” we’re running into true pipe capacity limits, and those limits often come with very hard stops. Typical business-class network connections don’t come close to handling high volumes, and you can forget about consumer-class connections handling them.

Beyond infrastructure issues, as engineers, the web app programming we’ve been doing over the past 15 years has taught us to build applications in a highly synchronous transactional manner. Because each HTTP transaction generally only lasts a second or so at most, it’s easy to digest and process many discrete chunks of data. However, the bastard stepchild of every HTTP lib’s “get()” routine that returns the complete result, is the “read()” routine that only gives you a poorly bounded chunk.

You would be shocked at the ratio of engineers who can’t build event-driven, asynchronous data processing applications, to those who can, yet this is a big part of this space. Lack of ecosystem knowledge around these kinds of programming primitives is a big problem. Many higher level abstractions exist for streaming HTTP apps, but they’re not industrial strength, and therefore you have to really know what’s going on to build your own.

Shifting back to infrastructure: Often the bigger issue plaguing the network itself is one of latency, not throughput. While data tends to move quickly once streaming connections are established, inevitable reconnects create gaps. The longer those connections take to stand up, the bigger the gaps. Run a traceroute to your favorite API and see how many hops you take. It’s not pretty. Latencies on the network are generally a function of router and gateway clutter, as our packets bounce across a dozen servers just to get to the main server and then back to the client.

How is Gnip addressing these issues?

Jud Valeski: On the infrastructure side, we are trying (successfully to-date) to use existing, relatively off the shelf, back plane network topologies in the cloud to build our systems. We live on EC2 Larges and XLs to ensure dedicated NICs in our clusters. That helps with the router and gateway clutter. We're also working with Amazon to ensure seamless connection upgrades as volumes increase. These are use cases they actually want to solve at a platform level, so our incentives are nicely aligned. We also play at the IP-stack level to ensure packet transmission is optimized for constant high-volume streams.

Once total volumes move past standard inbound and outbound connection capabilities, we will be offering dedicated interconnects. However, those come at a very steep price for us and our volume customers.

All of this leads me to my real answer: Trimming the fat.

While a sweet spot for us is certainly high-volume data consumers, there are many folks who don't want volume, they want coverage. Coverage of just the activities they care about; usually their customers' brands or products. We take on the challenge of digesting and processing the high volume on inbound, and distill the stream down to just the bits our coverage customers desire. You may need 100% of the activities that mention "good food," but that obviously isn't 100% of a publisher's fire hose. Processing high-velocity root streams on behalf of hundreds of customers without adversely impacting latency takes a lot of work. Today, that means good ol'-fashioned engineering.

What tools and infrastructure changes are needed to better handle big-data streaming?

Jud Valeski: "Big data" as we talk about it today has been slayed by lots of cool abstractions (e.g. Hadoop) that fit nicely into the way we think about the stack we all know and love. "Big streams," on the other hand, challenge the parallelization primitives folks have been solving for "big data." There's very little overlap, unfortunately. So, on the software solution side, better and more widely used frameworks are needed. Companies like BackType and Gnip pushing their current solutions onto

the network for open refinement would be an awesome step forward. I'm intrigued by the prospect of BackType's Storm project, and I'm looking forward to seeing more of it. More brains lead to better solutions.

We shouldn't be giving CPU and network latency injection a second thought, but we have to. The code I write to process bits as they come off the wire — quickly — should just “go fast,” regardless of its complexity. That's too hard today. It requires too much custom code.

On the infrastructure side of things, ISPs need to provide cheaper access to reliable fat pipes. If they don't, software will outpace their lack of innovation. To be clear, they don't get this and the software will lap them. You asked what I think we need, not what I think we'll actually get.

This interview was edited and condensed.

Chapter 2. Data Issues

Why the term “data science” is flawed but useful

Counterpoints to four common data science criticisms.

by Pete Warden

Mention “data science” to a lot of the high-profile people you might think practice it and you’re likely to see rolling eyes and shaking heads. It has taken me a while, but I’ve learned to love the term, despite my doubts. The key reason is that the rest of the world understands roughly what I mean when I use it. After years of stumbling through long-winded explanations about what I do, I can now say “I’m a data scientist” and move on. It is still an incredibly hazy definition, but my former descriptions left people confused as well, so this approach is no worse and at least saves time.

With that in mind, here are the arguments I’ve heard against the term, and why I don’t think they should stop its adoption.

It’s not a real science

I just finished reading “The Philosophical Breakfast Club,” the story of four Victorian friends who created the modern structure of science, as well as inventing the word “scientist.” I grew up with the idea that physics, chemistry and biology were the only real sciences and every other subject using the term was just stealing their clothes (“Anything that needs science in the name is not a real science”). The book shows that from the beginning the label was never restricted to just the hard experimental sciences. It was chosen to promote a disciplined approach to reasoning that relied on data rather than the poorly-supported logical deductions many contemporaries favored. Data science fits comfortably in this more open tradition.

It's an unnecessary label

To me, it's obvious that there has been a massive change in the landscape over the last few years. Data and the tools to process it are suddenly abundant and cheap. Thousands of people are exploiting this change, making things that would have been impossible or impractical before now, using a whole new set of techniques. We need a term to describe this movement, so we can create job ads, conferences, training and books that reach the right people. Those goals might sound very mundane, but without an agreed-upon term we just can't communicate.

The name doesn't even make sense

As a friend said, "show me a science that doesn't involve data." I hate the name myself, but I also know it could be a lot worse. Just look at other fields that suffer under terms like "new archaeology" (now more than 50 years old) or "modernist art" (pushing a century). I learned from teenage bands that the naming process is the most divisive part of any new venture, so my philosophy has always been to take the name you're given, and rely on time and hard work to give it the right associations. Apple and Microsoft (née Micro-soft) are terrible startup names by any objective measure, but they've earned their mindshare. People are calling what we're doing "data science," so lets accept that and focus on moving the subject forward.

There's no definition

This is probably the deepest objection, and the one with the most teeth. There is no widely accepted boundary for what's inside and outside of data science's scope. Is it just a faddish rebranding of statistics? I don't think so, but I also don't have a full definition. I believe that the recent abundance of data has sparked something new in the world, and when I look around I see people with shared characteristics who don't fit into traditional categories. These people tend to work beyond the narrow specialties that dominate the corporate and institutional world, handling everything from finding the data, processing it at scale, visualizing it and writing it up as a story. They also seem to start by looking at what the

data can tell them, and then picking interesting threads to follow, rather than the traditional scientist's approach of choosing the problem first and then finding data to shed light on it. I don't know what the eventual consensus will be on the limits of data science, but we're starting to see some outlines emerge.

Time for the community to rally

I'm betting a lot on the persistence of the term. If I'm wrong the Data Science Toolkit will end up sounding as dated as "surfing the information super-highway." I think data science, as a phrase, is here to stay though, whether we like it or not. That means we as a community can either step up and steer its future, or let others exploit its current name recognition and dilute it beyond usefulness. If we don't rally around a workable definition to replace the current vagueness, we'll have lost a powerful tool for explaining our work.

Why you can't really anonymize your data

It's time to accept and work within the limits of data anonymization.

by Pete Warden

One of the joys of the last few years has been the flood of real-world datasets being released by all sorts of organizations. These usually involve some record of individuals' activities, so to assuage privacy fears, the distributors will claim that any personally-identifying information (PII) has been stripped. The idea is that this makes it impossible to match any record with the person it's recording.

Something that my friend Arvind Narayanan has taught me, both with theoretical papers and repeated practical demonstrations, is that this anonymization process is an illusion. Precisely because there are now so many different public datasets to cross-reference, any set of records with a non-trivial amount of information on someone's actions has a good chance of matching identifiable public records. Arvind first demonstrated this when he and his fellow researcher took the "anonymous" dataset released as part of the first Netflix prize, and demonstrated how he could correlate the movie rentals listed with public IMDB reviews. That let them identify some named individuals, and then gave access to their complete rental histories. More recently, he and his collaborators used the same approach to win a Kaggle contest by matching the topography of the anonymized and a publicly crawled version of the social connections on Flickr. They were able to take two partial social graphs, and like piecing together a jigsaw puzzle, figure out fragments that matched and represented the same users in both.

All the known examples of this type of identification are from the research world — no commercial or malicious uses have yet come to light — but they prove that anonymization is not an absolute protection. In fact, it creates a false sense of security. Any dataset that has enough information on people to be interesting to researchers also has enough

information to be de-anonymized. This is important because I want to see our tools applied to problems that really matter in areas like health and crime. This means releasing detailed datasets on those areas to researchers, and those are bound to contain data more sensitive than movie rentals or photo logs. If just one of those sets is de-anonymized and causes a user backlash, we'll lose access to all of them.

So, what should we do? Accepting that anonymization is not a complete solution doesn't mean giving up, it just means we have to be smarter about our data releases. Below I outline four suggestions.

Keep the anonymization

Just because it's not totally reliable, don't stop stripping out PII. It's a good first step, and makes the reconstruction process much harder for any attacker.

Acknowledge there's a risk of de-anonymization

Don't make false promises to users about how anonymous their data is. Make the case to them that you're minimizing the risk and possible harm of any data leaks, sell them on the benefits (either for themselves or the wider world) and get their permission to go ahead. This is a painful slog, but the more organizations that take this approach, the easier it will be. A great model is Reddit, which asked their users to opt-in to sharing their data. They got a great response.

Limit the detail

Look at the records you're getting ready to open up to the world, and imagine that they can be linked back to named people. Are there parts of it that are more sensitive than others, and maybe less important to the sort of applications you have in mind? Can you aggregate multiple people together into cohorts that represent the average behavior of small groups?

Learn from the experts

There's many decades of experience of dealing with highly sensitive and personal data in sociology and economics departments across the globe. They've developed techniques that could prove useful to the emerging community of data scientists, such as subtle distortions of the information to prevent identification of individuals, or even the sort of locked-down clean-room conditions that are required to access detailed IRS data.

There's so much good that can be accomplished using open datasets, it would be a tragedy if we let this slip through our fingers with preventable errors. With a bit of care up front, and an acknowledgement of the challenges we face, I really believe we can deliver concrete benefits without destroying people's privacy.

Big data and the semantic web

At war, indifferent, or intimately connected?

by Edd Dumbill

On Quora, Gerald McCollum asked if big data and the semantic web were indifferent to each other, as there was little discussion of the semantic web topic at Strata this February.

My answer in brief is: big data's going to give the semantic web the massive amounts of metadata it needs to really get traction.

As the chair of the Strata conference, I see a vital link between big data and semantic web, and have my own roots in the semantic web world. Earlier this year however, the interaction was not yet of sufficient utility to make a strong connection in the conference agenda.

Google and the semantic web

A good example of the development of the relationship between big data and the semantic web is Google. Early on, Google search eschewed explicit use of semantics, preferring to infer a variety of signals in order to generate results. They used big data to create signals such as PageRank.

Now, as the search algorithms mature, Google's mission is to make their results ever more useful to users. To achieve this, their software must start to understand more about the actual world. Who's an author? What's a recipe? What do my friends find useful? So the connections between entities become more important. To achieve this Google is using data from initiatives such as schema.org, RDFa and microformats.

Google do not use these semantic web techniques to replace their search, but rather to augment it and make it more useful. To get all fancy pants about it: Google are starting to promote the information they gather toward being knowledge. They even renamed their search group as "Knowledge".

Metadata is hard: big data can help

Conventionally, semantic web systems generate metadata and identified entities explicitly, ie. by hand or as the output of database values. But as anybody who's tried to get users to do it will tell you, generating metadata is hard. This is part of why the full semantic web dream isn't yet realized. Analytical approaches take a different approach: surfacing and classifying the metadata from analysis of the actual content and data itself. (Freely exposing metadata is also controversial and risky, as open data advocates will attest.)

Once big data techniques have been successfully applied, you have identified entities and the connections between them. If you want to join that information up to the rest of the web, or to concepts outside of your system, you need a language in which to do that. You need to organize, exchange and reason about those entities. It's this framework that has been steadily built up over the last 15 years with the semantic web project.

To give an already widespread example: many data scientists use Wikipedia to help with entity resolution and disambiguation, using Wikipedia URLs to identify entities. This is a classic use of the most fundamental of semantic web technologies: the URI.

For Strata, as our New York series of conferences approaches, we will be starting to include a little more semantic web, but with a strict emphasis on utility.

Strata itself is not as much beholden to big data, as about being data-driven, and the ongoing consequences that has for technology, business and society.

Big data: Global good or zero-sum arms race?

It remains to be seen if big data will catalyze exponential growth.

by Jim Stogdill

Last month, Netezza CEO Jim Baum gave a talk at the GigaOM big data event. If I'm honest, I was checking my email and missed most of it, but I do remember tuning in just in time to hear him say something like "big data is going to have a huge economic impact."

I spend most of my days considering how the component pieces of this big data transformation will impact the corporate enterprise. Baum's comment got me thinking, though, about a more meta question: Is "big data" a key to some kind of industrial revolution reboot? Or, is it just going to be expensive table stakes for previously simple-to-understand businesses?

For 200-plus years the industrial revolution^[4] has been a kind of Moore's law of human productivity. Over that period our economic output per person has been growing like clockwork, and whatever you think of the various political -isms that sprung from industrialization, this march of productivity has pulled a lot of people out of poverty and is cause for the first sustained increase in wealth across human history.

But like Moore's law in a single core, our industrial revolution in advanced economies is kinda playing out. Our economy has been shifting for some time toward services that are proving to be impervious to order-of-magnitude productivity gains. The thousand-fold increases in productivity we saw on the farm and in the factory just don't seem likely to happen in health care and other service intensive sectors.

Of course our economy continues to grow, but at a rate that is staying just a skosh ahead of population growth. And since the top 1% are taking all of that (and perhaps more), for the first time in American history

parents are worrying that their kids won't have opportunities better than their own. Voila! There stems the populist anger that feeds the Tea Party.

That's the U.S.-centric view. Of course on a global basis there is tremendous growth as late-stage industrial revolution innovations are applied with vigor to developing economies. The 8% growth rates many countries are achieving will double their population's wealth every 10 years. But for the U.S., achieving growth requires parallelism. Of course, in this context we call it globalism and it means if we can't be more productive in one place, we have to take advantage of modern communications to do it in a bunch of other cheaper places. The problem is, after a few decades of those 8% overseas growth rates, there will be less comparative advantage for us to take advantage of and if we want continued economic growth, we really will need to find ways to be more productive.

So, that's why Baum's comment stuck in my head.

At the risk of way over generalizing, so far "big data" has mostly been about behavioral analysis to better target ads. Is that what Baum meant? That more effectively matching producer and consumer long tails through precision ad placement is going to fundamentally change the economy? That type of matching can promote economic activity, which is good, but I don't see the link to fundamentally improved productivity. If this kind of innovation pulls another tranche of the bell curve out of poverty it will do it by putting more people to work doing the same stuff, not by making our economy fundamentally more efficient.

When I heard "huge impact on the economy," my first thought was maybe it's just a throw-away comment. Maybe he just meant the economy as seen through the narrow lens of his company revenues. But then I tried to think about this on a deeper level: What's here that I haven't considered? Could he somehow be saying that this is a catalyst for the next big phase of productivity growth in our 200-year-old industrial revolution? Is this the industrial revolution equivalent of nanometer chip design, which starts the next decade of doubling? Is it the thing that gets the middle class growing again and eases all this populist anger?

Yeah, that might sound kind of absurd, but that's how my head works — a daily stream of ADHD-fueled big dreams immediately dashed on the rocks of reality.

(As an aside, half way through writing this I came across a prediction of the “wine and roses” we'll all experience with this “New Information Age.” Don't sweat the death of privacy, the surveillance state is highly unlikely.)

So, back to the question: Is big data an economic driver or just a must-have to be in the game?

As early as the 1950s it was obvious that robotic automation was going to fundamentally change manufacturing. As automobiles increasingly were built by robotic labor, the industry saw incredible productivity gains. The hours of human labor per automobile dropped by orders of magnitude over the next 30 years. Naturally, cars didn't just get cheaper, they also got more complex and feature-rich. But anyone could understand the return on capital of installing robotic lines. What's the return on capital look like for a Hadoop cluster?

It's worth noting that robots didn't just increase productivity, they also reshaped labor's relationship with management. If you're labor, competing with a robot sucks. This was presciently described by Norbert Wiener in his classic “The Human Use of Human Beings, Cybernetics and Society.” Of course we don't need history's warning to know that big data might have a dark side, too. If you don't see it now, you will when you download a new car stereo software version and it resets all your radio station presets based on Toyota's notion of people like you. Of course, for a car company to be as obnoxious as your software and search bar providers have long been, they have to learn as much about you as those software guys do, and that's weird. We aren't really used to the idea of a manufacturer knowing where we go and who we go there with.

There obviously are places where large-scale data and analysis will improve efficiencies and productivity. Particularly in areas like smart grid, where it will reduce the investment necessary in power plant

construction, or financial services, where it promises to help fight fraudulent transactions. What else? Are there big opportunities for order-of-magnitude productivity gains out there that come to mind? Or is most of the value created by this “new information age” going to be in some mushy upper region of Maslow’s hierarchy? A kind of middle class feel-good machine that remains completely irrelevant to the working poor dreaming of their first homes?

Norbert Weiner was concerned that automation-based productivity gains would disrupt the working man and woman’s living. He held that concern in the face of the obvious and compelling productivity gains that were sure to flow through to GDP as wealth.

As we enter the big data era of the information age and give up what’s left of our privacy, I’d like to think that it will be for more than a zero-sum game of musical chairs to decide the next winners.

[4] *For the purposes of this post, I’m treating the industrial age and the information age as two parts of one continuum.*

The truth about data: Once it's out there, it's hard to control

Jeff Jonas on data ownership, security concerns, and privacy trade offs.

by Jenn Webb

The amount of data being produced is increasing exponentially, which raises big questions about security and ownership. Do we need to be more concerned about the information many of us readily give out to join popular social networks, sign up for website community memberships, or subscribe to free online email? And what happens to that data once it's out there?

In a recent interview, Jeff Jonas (@JeffJonas), IBM distinguished engineer and a speaker at the O'Reilly Strata Online Conference, said consumers' willingness to give away their data is a concern, but it's perhaps secondary to the sheer number of data copies produced.

Our interview follows.

What is the current state of data security?

Jeff Jonas: A lot of data has been created, and a boatload more is on its way — we have seen nothing yet. Organizations now wonder how they are going to protect all this data — especially how to protect it from unintended disclosure. Healthcare providers, for example, are just as determined to prevent a “wicked leak” as anyone else. Just imagine the conversation between the CIO and the board trying to explain the risk of the enemy within — the “insider threat” — and the endless and ever-changing attack vectors.

I'm thinking a lot these days about data protection, ranging from reducing the number of copies of data to data anonymization to perpetual insider threat detection.

How are advancements in data gathering, analysis, and application affecting privacy, and should we be concerned?

Jeff Jonas: When organizations only collect what they need in order to conduct business, tell the consumer what they are collecting, why and how they are going to use it, and then use it this way, most would say “fair game.” This is all in line with Fair Information Practices (FIPs).

There continues to be some progress in the area of privacy-enhancing technology. For example, tamper-resistant audit logs, which are a way to record how a system was used that even the database administrator cannot alter. On the other hand, the trend that I see involves the willingness of consumers to give up all kinds of personal data in return for some benefit — free email or a fantastic social network site, for example.

While it is hard to not be concerned about what is happening to our privacy, I have to admit that for the most part technology advances are really delivering a lot of benefit to mankind.

What are the major issues surrounding data ownership?

Jeff Jonas: If users continue to give their data away because the benefits are irresistible, then there will be fewer battles, I suppose. The truth about data is that once it is out there, it’s hard to control.

I did a back of the envelope estimate a few years ago to estimate the number of copies a single piece of data may experience. Turns out the number is roughly the same as the number of licks it takes to get to the center of a Tootsie Pop — a play on an old TV commercial that basically translates to more than you can easily count.

A well-thought-out data backup strategy alone may create more than 100 copies. Then what about the operational data stores, data warehouses, data marts, secondary systems and their backups? Thousands of copies would not be uncommon. Even if a consumer thought they could own their data — which they can’t in many settings — how could they ever do anything to affect it?

Chapter 3. The Application of Data: Products and Processes

How the Library of Congress is building the Twitter archive

Checking in on the Library of Congress' Twitter archive, one year later.

by Audrey Watters

In April 2010, Twitter announced it was donating its entire archive of public tweets to the Library of Congress. Every tweet since Twitter's inception in 2006 would be preserved. The donation of the archive to the Library of Congress may have been in part a symbolic act, a recognition of the cultural significance of Twitter. Although several important historical moments had already been captured on Twitter when the announcement was made last year (the first tweet from space, for example, Barack Obama's first tweet as President, or news of Michael Jackson's death), since then our awareness of the significance of the communication channel has certainly grown.

That's led to a flood of inquiries to the Library of Congress about how and when researchers will be able to gain access to the Twitter archive. These research requests were perhaps heightened by some of the changes that Twitter has made to its API and firehose access.

But creating a Twitter archive is a major undertaking for the Library of Congress, and the process isn't as simple as merely cracking open a file for researchers to peruse. I spoke with Martha Anderson, the head of the library's National Digital Information Infrastructure and Preservation Program (NDIIP), and Leslie Johnston, the manager of the NDIIP's Technical Architecture Initiatives, about the challenges and opportunities of archiving digital data of this kind.

It's important to note that the Library of Congress is quite adept with the preservation of digital materials, as it's been handling these types of projects for more than a decade. The library has been archiving congressional and presidential campaign websites since 2000, for example, and it currently has more than 200 terabytes of web archives. It also has hundreds of terabytes of digitized newspapers, and petabytes of data from other sources, such as film archives and materials from the Folklife Center. So the Twitter archives fall within the purview of these sorts of digital preservation efforts, and in terms of the size of the archive, it is actually not too unwieldy.

Even with a long experience with archiving “born digital” content, Anderson says the Library of Congress “felt pretty brave about taking on Twitter.”

What makes the endeavor challenging, if not the size of the archive, is its composition: billions and billions and billions of tweets. When the donation was announced last year, users were creating about 50 million tweets per day. As of Twitter's fifth anniversary several months ago, that number has increased to about 140 million tweets per day. The data keeps coming too, and the Library of Congress has access to the Twitter stream via Gnip for both real-time and historical tweet data.

Each tweet is a JSON file, containing an immense amount of metadata in addition to the contents of the tweet itself: date and time, number of followers, account creation date, geodata, and so on. To add another layer of complexity, many tweets contain shortened URLs, and the Library of Congress is in discussions with many of these providers as well as with the Internet Archive and its 301works project to help resolve and map the links.

As it stands, Anderson and Johnston say they won't be crawling all these external sites and end-points, although Anderson says that in her “grand vision of the future” all of this data — not just from the Library of Congress but from all these different technological and cultural heritage institutions — would be linked. In the meantime, the Library of Congress won't be creating a catalog of all these tweets and all this data,

but they do want to be able to index the material so researchers can effectively search it.

This requires a significant technological undertaking on the part of the library in order to build the infrastructure necessary to handle inquiries, and specifically to handle the sorts of inquiries that researchers are clamoring for. Anderson and Johnston say that a cross-departmental team has been assembled at the library, and they're actively taking input from researchers to find out exactly what their needs for the material may be. Expectations also need to be set about exactly what the search parameters will be — this is a high-bandwidth, high-computing-power undertaking after all.

The project is still very much under construction, and the team is weighing a number of different open source technologies in order to build out the storage, management and querying of the Twitter archive. While the decision hasn't been made yet on which tools to use, the library is testing the following in various combinations: Hive, ElasticSearch, Pig, Elephant-bird, HBase, and Hadoop.

A pilot workshop is slated to run this summer with researchers who can help guide the Library of Congress in building out the archive and its accessibility. Anderson and Johnston say they expect an initial offering to be made available in four or five months. But even then, access to the Twitter archive will be restricted to "known researchers" who will need to go through the Library of Congress approval process to gain access to the data. Based on the sheer number of research requests, there are going to be plenty of scholars lined up to have a closer examination of this important cultural and technological archive.

Data journalism, data tools, and the newsroom stack

The 2011 Knight News Challenge winners illustrate data's ascendance in media and government.

by Alex Howard

MIT's recent Civic Media Conference and the latest batch of Knight News Challenge winners made one reality crystal clear: as a new era of technology-fueled transparency, innovation and open government dawns, it won't depend on any single CIO or federal program. It will be driven by a distributed community of media, nonprofits, academics and civic advocates focused on better outcomes, more informed communities and the new news, whatever form it is delivered in.

The themes that unite this class of Knight News Challenge winners were data journalism and platforms for civic connections. Each theme draws from central realities of the information ecosystems of today.

Newsrooms and citizens are confronted by unprecedented amounts of data and an expanded number of news sources, including a social web populated by our friends, family and colleagues. Newsrooms, the traditional hosts for information gathering and dissemination, are now part of a flattened environment for news, where news breaks first on social networks, is curated by a combination of professionals and amateurs, and then analyzed and synthesized into contextualized journalism.

Data journalism and data tools

In an age of information abundance, journalists and citizens alike all need better tools, whether we're curating the samizdat of the 21st century in the Middle East, like Andy Carvin, processing a late night data dump, or looking for the best way to visualize water quality to a nation of consumers. As we grapple with the consumption challenges presented by this deluge of data, new publishing platforms are also empowering us to

gather, refine, analyze and share data ourselves, turning it into information.

In this future of media, as Mathew Ingram wrote at GigaOm, big data meets journalism, in the same way that startups see data as an innovation engine, or civic developers see data as the fuel for applications. “The media industry is (hopefully) starting to understand that data can be useful for its purposes as well,” Ingram wrote. He continued:

... data and the tools to manipulate it are the modern equivalent of the microfiche libraries and envelopes full of newspaper clippings that used to make up the research arm of most media outlets. They are just tools, but as some of the winners of the Knight News Challenge have already shown, these new tools can produce information that might never have been found before through traditional means.

The Poynter Institute took note of the attention paid to data by the Knight Foundation as well. As Steve Myers reported, the Knight News Challenge gave \$1.5 million to projects that filter and examine data. The winners that relate to data journalism include:

- Overview, which is a tool to help journalists find stories in large amounts of data by cleaning, visualizing and interactively exploring large document and data sets. Associated Press data journalist Jonathan Stray called Overview the “Drupal of data visualization”.
- ScraperWiki, a favorite tool of civic coders at Code for America and elsewhere, enables anyone to collect, store and publish public data. With the Knight funding, the next version of ScraperWiki will be even more powerful.
- OpenBlock Rural will use the OpenBlock platform to partner with local governments and community newspapers to collect and publish data in North Carolina, including crime, real estate, school ratings and restaurant inspections.
- The PANDA Project will try to make research easier in the newsroom with a set of open source, web-based tools oriented at making it easier for journalists to use and analyze data.

I talked more with the AP's Jonathan Stray about data journalism and Overview at the MIT Civic Media in the video below. For an even deeper dive into his thinking on what journalists need in the age of big data, read his thoughts on "the editorial search engine."

The newsroom stack

With these investments in the future of journalism, more seeds have been planted to add to a "**newsroom stack**," to borrow a technical term familiar to Radar readers, combining a series of technologies for use in a given enterprise.

"I like the thought of it," said Brian Boyer, the project manager for PANDA, in an interview at the MIT Media Lab. "The newsroom stack could add up to the kit of tools that you ought to be using in your day to day reporting."

Boyer described how the flow of data might move from a spreadsheet (as a .CSV file) to Google Refine (for tidying, clustering, adding columns) to PANDA and then on to Overview or Fusion Tables or Many Eyes, for visualization. This is about "small pieces, loosely joined," he said. "I would rather build one really good small piece than one big project that does everything."

PANDA and Overview are squarely oriented at bread-and-butter issues for newsrooms in the age of big data. "It's a pain to search across datasets, but we also have this general newsroom content management issue," said Boyer. "The data stuck on your hard drive is sad data. Knowledge management isn't a sexy problem to solve, but it's a real business problem. People could be doing better reporting if they knew what was available. Data should be visible internally."

Boyer thinks the trends toward big data in media are pretty clear, and that he and other hacker journalists can help their colleagues to not only understand it but to thrive. "There's a lot more of it, with government releasing its stuff more rapidly," he said. "The city of Chicago is dropping two datasets a week right now. We're going for increased efficiency, to help people work faster and write better stories. Every

major news org in the country is hiring a news app developer right now. Or two. For smaller news organizations, it really works for them. Their data apps account for the majority of their traffic.”

Bridging the data divide

There’s some caution merited here. Big data is not a panacea to all things, in media or otherwise. Greg Borenstein explored some of these issues in his post on big data and cybernetics earlier this month. Short version: humans still matter in building human relationships and making sense of what matters, however good our personalized relevance engines for news become. Proponents of open data have to consider a complementary concern: digital literacy.

As Jesse Lichtenstein asserted “open data along isn’t enough,” following the thread of danah boyd’s “transparency is not enough talk at the 2010 Gov 2.0 Expo. Open data can empower the empowered.

To make open government data sing, infomediaries need to have time and resources. If we’re going to hope that citizens will draw their own conclusions from showing public data in real-time, we’ll need to educate them to be able to be critical thinkers. As Andy Carvin tweeted during the MIT Civic Media conference, “you need to be sure those people have high levels of digital literacy and media literacy.” There’s a data divide that has to be considered here, as Nick Clark Judd pointed out over at techPresident.

It looks like those concerns were at least partially factored into the judges’ decision on other Knight News Challenge winners. Spending Stories, from the Open Knowledge Foundation, is designed to add context to news stories based upon government data by connecting stories to the data used. Poderapedia will try to bring more transparency to Chile using data visualizations that draw upon a database of editorial and crowdsourced data. The State Decoded will try to make the law more user-friendly. The project has notable open government DNA: Waldo Jaquith’s work on OpenVirginia was aimed at providing an API for the Commonwealth.

There were citizen science and transparency projects alongside all of those data plays too, including:

- Public Laboratory, a tool kit and online community for grassroots data gathering and research that builds upon the success of Grassroots Mapping.
- NextDrop, a project to provide mobile access to information about water availability to residents of a city in India.

Given the recent story here at Radar on citizen science and crowdsourced radiation data, there's good reason to watch both of these projects evolve. And given research from the Pew Internet and Life Project on the role of the Internet as a platform for collective action, the effect of connecting like-minded citizens to one another through efforts like the Tiziano Project may prove far reaching.

The data analysis path is built on curiosity, followed by action

Why simplicity, empiricism, and DIY are keys to data analysis.

by Mac Slocum

A traditional view of data analysis involves precision, preparation, and methodical examination of defined datasets. Philipp Janert, author of “Data Analysis with Open Source Tools,” has a somewhat different perspective. Those traditional elements are still important, but Janert also thinks simplicity, experimentation, action, and natural curiosity all shape effective data work. He expands on these ideas in the following interview.

Is data analysis inherently complicated?

Philipp Janert: I observe a tendency to do something complicated and fancy; to bring in a statistical concept and other “sophisticated” stuff. The problem is that the sophisticated stuff isn’t that easy to understand.

Why not just look at the data set? Just look at it in an editor. Maybe you’ll see something. Or, draw some graphs. Graphs don’t require any sort of formal analytical training. These simple methods can be illuminating precisely because you don’t need anything complicated, and nothing is hidden.

Why do analysts shy from simplicity?

PJ: I often perceive a great sense of insecurity in my co-workers when it comes to math. Because of that, I get the sense people are trying to almost hide behind complicated methods.

The classic case for me is that usually within the first three minutes of a conversation, people start talking about standard deviations. It’s the one concept from classical statistics that everyone has heard of. But contextually, it’s not clear what “standard deviation” really means. Are they talking about what’s being measured by the standard deviation, namely the width of the distribution? Are they referring to one particular

measure and how it's being calculated? Do they mean the conclusions that can be drawn from standard deviations in the Normal case?

We need to keep it simple and not get sucked into abstract concepts that may or may not be fully understood.

What tool or method offers the best starting point for data analysis?

PJ: Start by plotting the data set. Plot all of the data points and look at them. Don't try to calculate indicator quantities or summary statistics. Just look at what you see in the plot. Almost anything worthwhile can be seen in a good graph.

Is there a defined career path for people who want to become data scientists?

PJ: The stunning development over the 12 months I was writing this book is that "big data" became the thing that's on everybody's mind. All of a sudden, people are really concerned about very large datasets. Of course, this seems to be mostly driven by the social networking phenomenon. But the question is: What do we do with that data?

I know that for my purposes, I never need big data. When I ask people what they do with big data, I've found that it's not what I would call "analysis" at all, because it does not involve the development of conceptual models. It does not involve the inductive/deductive cycle of scientific reasoning.

It falls into one of two camps. The first is reporting. For instance, if a company is being paid based on the number of pages they serve, then counting the number of served pages is important. The resulting log files tend to be huge, so that's technically big data. But it's a very straightforward counting and reporting game.

The other camp is what I consider "generalized search." These are scenarios like: If User A likes movies B, C, and D, what other specific movie might User A want? That's a form of searching because you're not actually trying to create a conceptual model of user behavior. You're comparing individual data points; you're trying to find the movie that has the greatest similarity to a very specific other set of predefined movies.

For this kind of generalized, exhaustive search, you need a lot of data because you look for the individual data points. But that's not really analysis as I understand it, either.

So coming back to your original question—is there a path to becoming a “data scientist?”—we need to first find out what data science might be. It will encompass different things: the kind of big data I mentioned; reporting and business intelligence; hopefully the kind of conceptual modeling that I do. But depending on what you're trying to accomplish, you could require very different skills.

For what I do—and this is really the only data analysis I can speak about with any sense of confidence—the most important skill is curiosity. This sounds a little tacky, but I mean it. Are you curious why the grass is green? Are you curious why is the sky blue? I'm talking about questions of this sort. These are representative of the inquisitive mind of a scientist. If you have that, you're in good shape and you can start anywhere.

Besides curiosity, are there other traits or skills that benefit data analysts?

PJ: You need experience with empirical work. And by that I mean someone who looks at the “idiot lights” on a router to make sure the cable is plugged in before they troubleshoot. We've all been in the situation where you reinstall the IP stack because you can't get network connectivity, and only later did you realize the router wasn't plugged in. These failures of empirical work are critical because empirical skills can be learned.

It's also nice, but not essential, to have taken a college math class and retained a bit. You should learn a programming language as well because you need to know how to manipulate data on your own. Any of the current scripting languages will do.

The last thing is that you need to actually do the work. Find a dataset that you're interested in and work on it. It doesn't have to be fancy, but you have to get started. You can't just sit there and expect it to happen. Experience and practice are really important.

It sounds like the “just start” mindset you find in the Maker/DIY community also applies to data. Is that right?

PJ: I don't know about other people, but I do this because it's fun. And that's a similar mentality to the Make space. They're more about creating something as opposed to understanding something, but the mentality is very much the same.

It's about curiosity followed by action. You look at the dataset and then go deeper to discover something. And this process isn't defined by tools. Personally, I'm interested in what somebody's trying to find rather than if they're using all the right statistical methods.

How data and analytics can improve education

George Siemens on the applications and challenges of education data.

by Audrey Watters

Schools have long amassed data: tracking grades, attendance, textbook purchases, test scores, cafeteria meals, and the like. But little has actually been done with this information — whether due to privacy issues or technical capacities — to enhance students' learning.

With the adoption of technology in more schools and with a push for more open government data, there are clearly a lot of opportunities for better data gathering and analysis in education. But what will that look like? It's a politically charged question, no doubt, as some states are turning to things like standardized test score data in order to gauge teacher effectiveness and, in turn, retention and promotion.

I asked education theorist George Siemens, from the Technology Enhanced Knowledge Research Institute at Athabasca University, about the possibilities and challenges for data, teaching, and learning.

Our interview follows.

What kinds of data have schools traditionally tracked?

George Siemens: Schools and universities have long tracked a broad range of learner data — often drawn from applications (universities) or enrollment forms (schools). This data includes any combination of: location, previous learning activities, health concerns (physical and emotional/mental), attendance, grades, socio-economic data (parental income), parental status, and so on. Most universities will store and aggregate this data under the umbrella of institutional statistics.

Privacy laws differ from country to country, but generally will prohibit academics from accessing data that is not relevant to a particular class, course, or program. Unfortunately, most schools and universities do very

little with this wealth of data, other than possibly producing an annual institutional profile report. Even a simple analysis of existing institutional data could raise the profile of potential at-risk students or reveal attendance or assignment submission patterns that indicate the need for additional support.

What new types of educational data can now be captured and mined?

George Siemens: In terms of learning analytics or educational data-mining, the growing externalization of learning activity (i.e. capturing how learners interact with content and the discourse they have around learning materials as well as the social networks they form in the process) is driven by the increased attention to online learning. For example, a learning management system like Moodle or Desire2Learn captures a significant amount of data, including time spent on a resource, frequency of posting, number of logins, etc. This data is fairly similar to what Google Analytics or Piwik collects regarding website traffic. A new generation of tools, such as SNAPP, uses this data to analyze social networks, degrees of connectivity, and peripheral learners. Discourse analysis tools, such as those being developed at the Knowledge Media Institute at the Open University, UK, are also effective at evaluating the qualitative attributes of discourse and discussions and rate each learner's contributions by depth and substance in relation to the topic of discussion.

An area of data gathering that universities and schools are largely overlooking relates to the distributed social interactions learners engage in on a daily basis through Facebook, blogs, Twitter, and similar tools. Of course, privacy issues are significant here. However, as we are researching at Athabasca University, social networks can provide valuable insight into how connected learners are to each other and to the university. Potential models are already being developed on the web that would translate well to school settings. For example, Klout measures influence within a network and Radian6 tracks discussions in distributed networks.

The existing data gathering in schools and universities pales in comparison to the value of data mining and learning analytics opportunities that exist in the distributed social and informational networks that we all participate in on a daily basis. It is here, I think, that most of the novel insights on learning and knowledge growth will occur. When we interact in a learning management system (LMS), we do so purposefully — to learn or to complete an assignment. Our interaction in distributed systems is more “authentic” and can yield novel insights into how we are connected, our sentiments, and our needs in relation to learning success. The challenge, of course, is how to balance concerns of the Hawthorne effect with privacy.

Discussions about data ownership and privacy lag well behind what is happening in learning analytics. Who owns learner-produced data? Who owns the analysis of that data? Who gets to see the results of analysis? How much should learners know about the data being collected and analyzed?

I believe that learners should have access to the same dashboard for analytics that educators and institutions see. Analytics can be a powerful tool in learner motivation — how do I compare to others in this class? How am I doing against the progress goals that I set? If data and analytics are going to be used for decision making in teaching and learning, then we need to have important conversations about who sees what and what are the power structures created by the rules we impose on data and analytics access.

How can analytics change education?

George Siemens: Education is, today at least, a black box. Society invests significantly in primary, secondary, and higher education. Unfortunately, we don't really know how our inputs influence or produce outputs. We don't know, precisely, which academic practices need to be curbed and which need to be encouraged. We are essentially swatting flies with a sledgehammer and doing a fair amount of peripheral damage.

Learning analytics are a foundational tool for informed change in education. Over the past decade, calls for educational reform have

increased, but very little is understood about how the system of education will be impacted by the proposed reforms. I sometimes fear that the solution being proposed to what ails education will be worse than the current problem. We need a means, a foundation, on which to base reform activities. In the corporate sector, business intelligence serves this “decision foundation” role. In education, I believe learning analytics will serve this role. Once we better understand the learning process — the inputs, the outputs, the factors that contribute to learner success — then we can start to make informed decisions that are supported by evidence.

However, we have to walk a fine line in the use of learning analytics. On the one hand, analytics can provide valuable insight into the factors that influence learners’ success (time on task, attendance, frequency of logins, position within a social network, frequency of contact with faculty members or teachers). Peripheral data analysis could include the use of physical services in a school or university: access to library resources and learning help services. On the other hand, analytics can’t capture the softer elements of learning, such as the motivating encouragement from a teacher and the value of informal social interactions. In any assessment system, whether standardized testing or learning analytics, there is a real danger that the target becomes the object of learning, rather than the assessment of learning.

With that as a caveat, I believe learning analytics can provide dramatic, structural change in education. For example, today, our learning content is created in advance of the learners taking a course in the form of curriculum like textbooks. This process is terribly inefficient. Each learner has differing levels of knowledge when they start a course. An intelligent curriculum should adjust and adapt to the needs of each learner. We don’t need one course for 30 learners; each learner should have her own course based on her life experiences, learning pace, and familiarity with the topic. The content in the courses that we take should be as adaptive, flexible, and continually updated. The black box of education needs to be opened and adapted to the requirements of each individual learner.

In terms of evaluation of learners, assessment should be in-process, not at the conclusion of a course in the form of an exam or a test. Let's say we develop semantically-defined learning materials and ways to automatically compare learner-produced artifacts (in discussions, texts, papers) to the knowledge structure of a field. Our knowledge profile could then reflect how we compare to the knowledge architecture of a domain — i.e. “you are 64% on your way to being a psychologist” or “you are 38% on your way to being a statistician.” Basically, evaluation should be done based on a complete profile of an individual, not only the individual in relation to a narrowly defined subject area.

Programs of study should also include non-school-related learning (prior learning assessment). A student that volunteers with a local charity or a student that plays sports outside of school is acquiring skills and knowledge that is currently ignored by the school system. “Whole-person analytics” is required where we move beyond the micro-focus of exams. For students that return to university mid-career to gain additional qualifications, recognition for non-academic learning is particularly important.

Much of the current focus on analytics relates to reducing attrition or student dropouts. This is the low-hanging fruit of analytics. An analysis of the signals learners generate (or fail to — such as when they don't login to a course) can provide early indications of which students are at risk for dropping out. By recognizing these students and offering early interventions, schools can reduce dropouts dramatically.

All of this is to say that learning analytics serve as a foundation for informed change in education, altering how schools and universities create curriculum, deliver it, assess student learning, provide learning support, and even allocate resources.

What technologies are behind learning analytics?

George Siemens: Some of the developments in learning analytics track the development of the web as a whole — including the use of recommender systems, social network analysis, personalization, and adaptive content. We are at an exciting cross-over point between

innovations in the technology space and research in university research labs. Language recognition, artificial intelligence, machine learning, neural networks, and related concepts are being combined with the growth of social network services, collaborative learning, and participatory pedagogy.

The combination of technical and social innovations in learning offers huge potential for a better, more effective learning model. Together with Stephen Downes and Dave Cormier, I've experimented with "massive open online courses" over the past four years. This experimentation has resulted in software that we've developed to encourage distributed learning, while still providing a loose level of aggregation that enables analytics. Tools like Open Study take a similar approach: decentralized learning, centralized analytics. Companies like Grockit and Knewton are creating personalized adaptive learning platforms. Not to be outdone, traditional publishers like Pearson and McGraw-Hill are investing heavily in adaptive learning content and are starting to partner with universities and schools to deliver the content and even evaluate learner performance. Learning management system providers (such as Desire2Learn and Blackboard) are actively building analytics options into their offerings.

Essentially, in order for learning analytics to have a broad impact in education, the focus needs to move well beyond basic analytics techniques such as those found in Google Analytics. An integrated learning and knowledge model is required where the learning content is adaptive, prior learning is included in assessment, and learning resources are provided in various contexts (e.g. "in class today you studied Ancient Roman laws, two blocks from where you are now, a museum is holding a special exhibit on Roman culture"). The profile of the learner, not pre-planned content, needs to drive curriculum and learning opportunities.

What are the major obstacles facing education data and analytics?

George Siemens: In spite of the enormous potential they hold to improve education, learning analytics are not without concerns. Privacy for learners and teachers is a critical issue. While I see analytics as a

means to improve learner success, opportunities exist to use analytics to evaluate and critique the performance of teachers. Data access and ownership are equally important issues: who should be able to see the analysis that schools perform on learners? Other concerns relate to error-correction in analytics. If educators rely heavily on analytics, effort should be devoted to evaluating the analytics models and understanding in which contexts those analytics are not valid.

With regard to the adoption of learning analytics, now is an exceptionally practical time to explore analytics. The complex challenges that schools and universities face can, at least partially, be illuminated through analytics applications.

Data science is a pipeline between academic disciplines

Drew Conway on how data science intersects with research and the social sciences.

by Audrey Watters

We talk a lot about the ways in which data science affects various businesses, organizations, and professions, but how are we actually preparing future data scientists? What training, if any, do university students get in this area? The answer may be obvious if students focus on math, statistics or hard science majors, but what about other disciplines?

I recently spoke with Drew Conway (@drewconway) about data science and academia, particularly in regards to social sciences. Conway, a PhD candidate in political science at New York University, will expand on some of these topics during a session at next month's Strata Conference in New York.

Our interview follows.

How has the work of academia — particularly political science — been affected by technology, open data, and open source?

Drew Conway: There are fundamentally two separate questions in here, so I will try to address both of them. First is the question of how academic research has changed as a result of these technologies. And for my part, I can only really speak for how they have affected social science research. The open data movement has impacted research most notably in compressing the amount of time a researcher goes from the moment of inception (“hmm, that would be interesting to look at!”) to actually looking at data and searching for interesting patterns. This is especially true of the open data movement happening at the local, state and federal government levels.

Only a few years ago, the task of identifying, collecting, and normalizing these data would have taken months, if not years. This meant that a

researcher could have spent all of that time and effort only to find out that their hypothesis was wrong and that — in fact — there was nothing to be found in a given dataset. The richness of data made available through open data allows for a much more rapid research cycle, and hopefully a greater breadth of topics being researched.

Open source has also had a tremendous impact on how academics do research. First, open source tools for performing statistical analysis, such as R and Python, have robust communities around them. Academics can develop and share code within their niche research area, and as a result the entire community benefits from their effort. Moreover, the philosophy of open source has started to enter into the framework of research. That is, academics are becoming much more open to the idea of sharing data and code at early stages of a research project. Also, many journals in the social sciences are now requiring that authors provide replication code and data.

The second piece of the question is how these technologies affect the dissemination of research. In this case blogs have become the de facto source for early access to new research, or scientific debate. In my own discipline, The Monkey Cage is most political scientists' first source for new research. What is fantastic about the Monkey Cage, and other academic blogs, is that they are not only read by other academics. Journalists, policy makers, and engaged citizens can also interact with academics in this way — something that was not possible before these academic blogs became mainstream.

Let's sidestep the history of the discipline and debates about what constitutes a hard or soft science. But as its name suggests, "political science" has long been interested in models, statistics, quantifiable data and so on. Has the discipline been affected by the rise of data science and big data?

Drew Conway: The impact of big data has been slow, but there are a few champions who are doing really interesting work. Political science, at its core, is most interested in understanding how people collectively make decisions, and as researchers we attempt to build models and

collect data to that end. As such, the massive data on social interactions being generated by social media services like Facebook and Twitter present unprecedented opportunities for research.

While some academics have been able to leverage this data for interesting work, there seems to be a clash between these services' terms of service and with the desire for scientists to collect data and generate reproducible findings from this data. I wrote about my own experience using Twitter data for research, but there are many others researchers from all disciplines that have run into similar problems.

With respect to how academics have been impacted by data science, I think the impact has mostly flowed in the other direction. One major component of data science is the ability to extract insight from data using tools from math, statistics and computer science. Most of this is informed by the work of academics, and not the other way around. That said, as more academic researchers become interested in examining large-scale datasets (on the order of Twitter or Facebook), many of the technical skills of data science will have to be acquired by academics.

How does data science change the work of the grad student — in terms of necessary skills but also in terms of access to information/informants?

Drew Conway: Unfortunately, having sophisticated technical skills, i.e., those of a data scientist, are still undervalued in academia. Being involved in open-source projects, or producing statistical software is not something that will help a graduate student land a high-profile academic job, or help a young faculty member get tenure. Publications are still the currency of success, and that — as I mentioned — clashes with the data-sharing policies of many large social media services.

Graduate students and faculty do themselves a disservice by not actively staying technically relevant. As so much more data gets pushed into the open, I believe basic data hacking skills — scraping, cleaning, and visualization — will be prerequisites to any academic research project. But, then again, I've always been a weird academic, double majoring in computer science and political science as an undergrad

How does the rise of data science and its spread beyond the realm of math and statistics change the world of technology, either from an academic or entrepreneurial perspective?

Drew Conway: From an entrepreneurial perspective I think it has dramatically changed the way new businesses think about building a team. Whether it is at Strata, or any of the other conferences in the same vein, you will see a glut of job openings or panels on how to “build a data team.” At present, people who have the blend of skills I associate with data science — hacking, math/stats, and substantive expertise — are a rare commodity. This dearth of talent, however, will be short-lived.

I see in my undergrads many more students who grew up with data and computing as ubiquitous parts of their lives. They’re interested in pursuing routes of study that provide them with data science skills, both in terms of technical competence, and also in creative outlets such as interactive design.

How does “human subjects compliance” work when you’re talking about “data” versus “people” — that’s an odd distinction, of course, and an inaccurate one at that. But I’m curious if some of the rules and regulations that govern research on humans account for research on humans’ data.

Drew Conway: I think it is an excellent question, and one that academe is still struggling to deal with. In some sense, mining social data that is freely available on the Internet provides researchers a way to sidestep traditional IRB regulation. I don’t think there’s anything ethically questionable about recording observations that are freely made public. That’s akin to observing the meanderings of people in a park.

Where things get interesting is when researchers use crowd sourcing technology, like Mechanical Turk, as a survey mechanism. Here, this is much more of a gray area. I suppose, technically, the Amazon terms of services covers researchers, but ethically this is something that would seem to me to fall within the scope of an IRB. Unfortunately, the likely outcome is that institutions won’t attempt to understand the difference until some problem arises.

This interview was edited and condensed.

Big data and open source unlock genetic secrets

Charlie Quinn is mixing data to advance genetic discovery.

by Alex Howard

The world is experiencing an unprecedented data deluge, a reality that my colleague Edd Dumbill described as another “industrial revolution” at February’s Strata Conference. Many sectors of the global economy are waking up to the need to use data as a strategic resource, whether in media, medicine, or moving trucks. Open data has been a major focus of Gov 2.0, as federal and state governments move forward with creating new online platforms for open government data.

The explosion of data requires new tools and management strategies. These new approaches include more than technical evolution, as a recent conversation with Charlie Quinn, director of data integration technologies at the Benaroya Research Institute, revealed: they involve cultural changes that create greater value by sharing data between institutions. In Quinn’s field, genomics, big data is far from a buzzword, with scanned sequences now rating on the terabyte scale.

In the interview below, Quinn shares insights about applying open source to data management and combining public data with experimental data. You can hear more about open data and open source in advancing personalized medicine from Quinn at the upcoming OSCON Conference.

How did you become involved in data science?

Charlie Quinn: I got into the field through a friend of mine. I had been doing data mining for fraud on credit cards and the principal investigator, who I work with now, was going to work in Texas. We had a novel idea that to build the tools for researchers, we should hire software people. What had happened in the past was you had bioinformaticians writing scripts. They found the programs that they needed did about 80% of what they wanted, and they had a hard time

gaining the last 20%. So we had had a talk way back when saying, "if you really want proper software tools, you ought to hire software people to build them for you." He called my boss to come on down and take a look. I did, and the rest is history.

You've said that there's a "data explosion" in genomics research. What do you mean? What does this mean for your field?

Charlie Quinn: It's like the difference between analog and digital technology. The amount of data you'd have with analog is still substantial, but as we move toward digital, it grows exponentially. If we're looking at technology in gene expression values, which is what we've been focusing on in genomics, it's about a gigabyte per scan. As we move into doing targeted RNA sequencing, or even high frequency sequencing, if you take the raw output from the sequence, you're looking at terabytes per scan. It's orders of magnitude more data.

What that means from a practical perspective is there's more data being generated than just for your request. There's more data being generated than a single researcher could possibly ever hope to get their head wrapped around. Where the data explosion becomes interesting is how we engage researchers to take data they're generating and share it with others, so that we can reuse data, and other people might be able to find something interesting in it.

What are the tools you're using to organize and make sense of all that data?

Charlie Quinn: A lot of it's been homegrown so far, which is a bit of an issue as you start to integrate with other organizations because everybody seems to have their own homegrown system. There's an open source group in Seattle called Lab Key, which a lot of people have started to use. We're taking another look at them to see if we might be able to use some of their technology to help us move forward in organizing the backend. A lot of this is so new. It's hard to keep up with where we're at and quite often, we're outpacing it. It's a question of homegrown and integrating with other applications as we can.

How does open source relate to that work?

Charlie Quinn: We try and use open source as much as we can. We try and contribute back where we can. We haven't been contributing back anywhere near as much as we'd like to, but we're going to try and get into that more.

We're huge proponents not only of open source, but of open data. What we've been doing is going around and trying to convince people that we understand they have to keep data private up to a certain point, but let's try and release as much data as we can as early as we can.

When we go back to talking about the explosion of data, if we're looking at Gene X and we happen to see something that might be interesting on Y or Z, we can post a quick discovery note or a short blurb. In that way, you're trying to push ideas out and take the data behind those ideas and make it public. That's where I think we're going to get traction: trying to share data earlier rather than later.

At OSCON, you'll talk about how experimental data combines with public data. When did you start folding the two together?

Charlie Quinn: We've been playing with it for a while. What we're hoping to do is make more of it public, now that we're getting the institutional support for it. Years ago, we went and indexed all of the abstracts at Pubnet by gene so that when people went to a text engine, you could type in your query and you would get a list of genes, as opposed to a list of articles. That helped researchers find what they were looking for — and that's just leveraging openly available data. Now, with NIH's mandate for more people to publish their results back into repositories, we're downloading that data and combining it with the data we have internally. Now, as we go across a project or across a disease trying to find how a gene is acting or how a protein is acting, it's just giving us a bigger dataset to work with.

What are some of the challenges you've encountered in your work?

Charlie Quinn: The issues we've had are with the quality of the datasets in the public repositories. You need to hire a curator to validate if the data is going to be usable or not, to make sure it's comparable to the data that we want to use it with.

What's the future of open data in research and personalized medicine?

Charlie Quinn: We're going to be seeing multiple tiers of data sharing. In the long run, you've going to have very well curated public repositories of data. We're a fair ways away from there in reality because there's still a lot of inertia against doing that within the research community. The half-step to get there will be large project consortiums where we start sharing data inter-institutionally. As people get more comfortable with that, we'll be able to open it up to a wider audience.

This interview was edited and condensed.

Visualization deconstructed: Mapping Facebook's friendships

A deep look at Paul Butler's popular Facebook visualization.

by Sébastien Pierre

In the first post in Radar's new "visualization deconstructed" series, I talked about how data visualization originated from cartography (which some now just call "mapping"). Cartography initially focused on mapping physical spaces, but at the end of the 20th century we created and discovered new spaces that were made possible by the Internet. By abstracting away the constraints of the physical space, social networks such as Facebook emerged and opened up new territories, where topology is primarily defined by the social fabric rather than physical space. But is this fabric completely de-correlated from the physical space?

Mapping Facebook's friendships

Last December, Paul Butler, an intern on Facebook's data infrastructure engineering team, posted a visualization that examined a subset of the relations between Facebook users. Users were positioned in their respective cities and arcs denoted friendships.

Paul extracted the data and started playing with it. As he put it:

Visualizing data is like photography. Instead of starting with a blank canvas, you manipulate the lens used to present the data from a certain angle.

There is definitely discovery involved in the process of creating a visualization, where by giving visual attributes to otherwise invisible data, you create a form for data to embody.



The most striking discovery that Paul made while creating his visualization was the unraveling of a very detailed map of the world, including the shapes of the continents (remember that only lines representing relationships are drawn).

If you compare the Facebook visualization with NASA's world at night pictures, you can see how close the two maps are, except for Russia and parts of China. It seems that Facebook has a big growth opportunity in these regions!



So let's have a look at Paul's visualization:

- A complex network of arcs and lines does a great job communicating the notions of human activity and organic social fabric.
- The choice of color palette works very well, as it immediately makes us think about night shots of earth, where the light of the city makes human activity visible. The color contrast is well balanced, so that we don't see too much blurring or bleeding of colors.
- Choosing to draw only lines and arcs makes the visualization very interesting, as at first sight, we would think that the outlines of continents and the cities have been pre-drawn. Instead, they emerge from the drawing of arcs representing friendships between people in different cities, and we can make the interesting discovery of a possible correlation between physical location and social friendships on the Internet.

Overall, this is a great visualization that had a lot of success last December, being mentioned in numerous blogs and liked by more than 2,000 people on Facebook. However, I can see a couple ways to improve it and open up new possibilities:

- **Play with the color scale**—By using a less linear gradient as a color scale, or by using more than two colors, some other patterns may

emerge. For instance, by using a clearer cut-off in the gradient, we could better see relations with a weight above a specific threshold. Also, using more than one color in the gradient might reveal the predominance of one color over another in specific regions. Again, it's something to try, and we'll probably lose some of the graphic appeal in favor of (perhaps) more insights into the data.

- **Play with the drawing of the lines**—Because the lines are spread all over the map, it's a little difficult to identify “streams” of lines that all flow in the same direction. It would be interesting to draw the lines in three parts, where the middle part would be shared by many lines, creating “pipelines” of relationships from one region to another. Of course, this would require a lot of experimentation and it might not even be possible with the tools used to draw the visualization.
- **Use a different reference to position cities**—Cities in the visualization are positioned using their geographical position, but there are other ways they could be placed. For instance, we could position them on a grid, ordered by their population, or GDP. What kind of patterns and trends would emerge by changing this perspective ?

Static requires storytelling

In last week's post, I looked at an interactive visualization, where users can explore the data and its different representations. With the Facebook data, we have a static visualization where we can only look, not touch — it's like gazing at the stars.

Although a static visualization has the potential to evolve into an interactive visualization, I think creating a static image involves a little bit more care. Interactive visualizations can be used as exploration tools, but static visualizations need to present insight the data explorer had when creating the visualization. It has to tell a story to be interesting.

Data science democratized

With new tools arriving, data science may soon be in the hands of non-programmers.

by Mac Slocum

I am not a data scientist. Nor am I a programmer. I've got an inclination toward technology, but my core skill set very much resides in the humanities domain.

I offer this biographical sketch up front because I think I have a lot in common with the people who work around and near tech spaces: academics, business users, entertainment professionals, editors, writers, producers, etc. The interesting thing about data science—and the reason why I'm glad Mike Loukides wrote "What is data science?"—is that vast stores of data have relevance to all sorts of folks, including people like me who lack a pure technical pedigree.

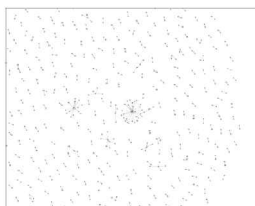
Data science's democratizing moment will come when its associated tools can be picked up by tech-savvy non-programmers. I'm thinking of the HTML coders and the Excel power users: the people who aren't full-fledged mechanics, but they're skilled enough to pop the hood and change their own oil.

I'm encouraged because that democratizing moment is close. I saw a demo recently that connects a web-based spreadsheet with huge data stores and cloud infrastructure. This type of system—and I'm sure there are many others in the pipeline—takes a process that once had immense technical and financial barriers and makes it almost as easy as phpMyAdmin. That's an important step. Within a year or two, I expect to see further usability improvements in these tools. A data science dashboard that mimics Google Analytics can't be far off.

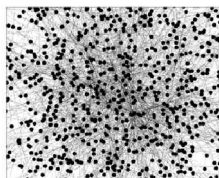
During that demo, Datameer CTO Stefan Groschupf told me about a fun Twitter inquiry he instigated. Groschupf had previously gathered around 45 million tweets and fed them into EC2. Later, over the course of a two-beer evening, Groschupf poked at that data to see if any interesting

patterns turned up when comparing two vastly different hashtags (#justinbieber vs. #teaparty). He used his company's system to parse the data, then he fed results through a free visualization tool.

Here's the #justinbieber cluster:



And here's the #teaparty cluster:



As you can see, the #teaparty folks are far more connected than their distant #justinbieber cousins. That's interesting, but not really surprising. The political world has more connective tissue than of-the-moment entertainment.

But that specific conclusion isn't what's important here. Even if your end-point is inevitable, a data-driven conversation has more power and resonance than an anecdotal observation. Groschupf didn't tell me the Tea Party movement is more connected. He *showed* me.

Significant implications emerge when you can bounce a question, even an innocuous one, against a huge storehouse of data. If someone like me can plug questions into a system and have it do the same kind of processing once reserved for a skilled minority, that will inspire me to ask a lot more questions. It'll inspire a lot of other people to ask questions, too. And some of those questions might even be important.

That's a big deal. Myself and others may never become full-fledged data scientists, but having access to easy-to-use data tools will get people thinking and exploring in all sorts of domains.

Chapter 4. The Business of Data

There's no such thing as big data

Even if you have petabytes of data, you still need to know how to ask the right questions to apply it.

by Alistair Croll

"You know," said a good friend of mine last week, "there's really no such thing as big data."

I sighed a bit inside. In the past few years, cloud computing critics have said similar things: that clouds are nothing new, that they're just mainframes, that they're just painting old technologies with a cloud brush to help sales. I'm wary of this sort of techno-Luddism. But this person is sharp, and not usually prone to verbal linkbait, so I dug deeper.

He's a ridiculously heavy traveler, racking up hundreds of thousands of miles in the air each year. He's the kind of flier airlines dream of: loyal, well-heeled, and prone to last-minute, business-class trips. He's exactly the kind of person an airline needs to court aggressively, one who represents a disproportionately large amount of revenues. He's an outlier of the best kind. He'd been a top-ranked passenger with United Airlines for nearly a decade, using their Mileage Plus program for everything from hotels to car rentals.

And then his company was acquired.

The acquiring firm had a contractual relationship with American Airlines, a competitor of United with a completely separate loyalty program. My friend's air travel on United and its partner airlines dropped to nearly nothing.

He continued to book hotels in Shanghai, rent cars in Barcelona, and buy meals in Tahiti, and every one of those transactions was tied to his

loyalty program with United. So the airline knew he was traveling—just not with them.

Astonishingly, nobody ever called him to inquire about why he'd stopped flying with them. As a result, he's far less loyal than he was. But more importantly, United has lost a huge opportunity to try to win over a large company's business, with a passionate and motivated inside advocate.

And this was his point about big data: *that given how much traditional companies put it to work, it might as well not exist.* Companies have countless ways they might use the treasure troves of data they have on us. Yet all of this data lies buried, sitting in silos. It seldom sees the light of day.

When a company does put data to use, it's usually a disruptive startup. Zappos and customer service. Amazon and retailing. Craigslist and classified ads. Zillow and house purchases. LinkedIn and recruiting. eBay and payments. Ryanair and air travel. One by one, industry incumbents are withering under the harsh light of data.

Big data and the innovator's dilemma

Large companies with entrenched business models tend to cling to their buggy-whips. They have a hard time breaking their own business models, as Clay Christensen so clearly stated in "The Innovator's Dilemma," but it's too easy to point the finger at simple complacency.

Early-stage companies have a second advantage over more established ones: they can ask for forgiveness instead of permission. Because they have less to lose, they can make risky bets. In the early days of PayPal, the company could skirt regulations more easily than Visa or Mastercard, because it had far less to fear if it was shut down. This helped it gain marketshare while established credit-card companies were busy with paperwork.

The real problem is one of asking the right questions.

At a big data conference run by *The Economist* this spring, one of the speakers made a great point: **Archimedes had taken baths before.**

(Quick historical recap: In an almost certainly apocryphal tale, Hiero of Syracuse had asked Archimedes to devise a way of measuring density, an indicator of purity, in irregularly shaped objects like gold crowns. Archimedes realized that the level of water in a bath changed as he climbed in, making it an indicator of volume. Eureka!)

The speaker's point was this: it was the *question* that prompted Archimedes' realization.

Small, agile startups disrupt entire industries because they look at traditional problems with a new perspective. They're fearless, because they have less to lose. But big, entrenched incumbents should still be able to compete, because they have massive amounts of data about their customers, their products, their employees, and their competitors. They fail because often they just don't know how to ask the right questions.

In a recent study, McKinsey found that by 2018, the U.S. will face a shortage of 1.5 million managers who are fluent in data-based decision making. It's a lesson not lost on leading business schools: several of them are introducing business courses in analytics.

Ultimately, this is what my friend's airline example underscores. It takes an employee, deciding that the loss of high-value customers is important, to run a query of all their data and find him, and then turn that into a business advantage. Without the right questions, there really is no such thing as big data—and today, it's the upstarts that are asking all the good questions.

When it comes to big data, you either use it or lose.

This is what we're hoping to explore at Strata JumpStart in New York next month. Rather than taking a vertical look at a particular industry, we're looking at the basics of business administration through a big data lens. We'll be looking at apply big data to HR, strategic planning, risk management, competitive analysis, supply chain management, and so on. In a world flooded by too much data and too many answers, tomorrow's business leaders need to learn how to ask the right questions.

Building data startups: Fast, big, and focused

Low costs and cloud tools are empowering new data startups.

by Michael Driscoll

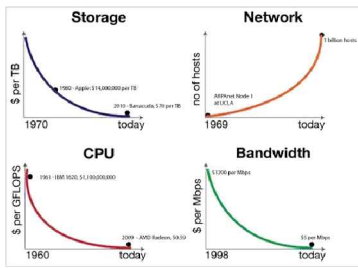
This is a written follow-up to a talk presented at a recent Strata online event.

A new breed of startup is emerging, built to take advantage of the rising tides of data across a variety of verticals and the maturing ecosystem of tools for its large-scale analysis.

These are data startups, and they are the sumo wrestlers on the startup stage. The weight of data is a source of their competitive advantage. But like their sumo mentors, size alone is not enough. The most successful of data startups must be fast (with data), big (with analytics), and focused (with services).

Setting the stage: The attack of the exponentials

The question of why this style of startup is arising today, versus a decade ago, owes to a confluence of forces that I call the Attack of the Exponentials. In short, over the past five decades, the cost of storage, CPU, and bandwidth has been exponentially dropping, while network access has exponentially increased. In 1980, a terabyte of disk storage cost \$14 million dollars. Today, it's at \$30 and dropping. Classes of data that were previously economically unviable to store and mine, such as machine-generated log files, now represent prospects for profit.



At the same time, these technological forces are not symmetric: CPU and storage costs have fallen faster than that of network and disk IO. Thus data is heavy; it gravitates toward centers of storage and compute power in proportion to its mass. Migration to the cloud is the manifest destiny for big data, and the cloud is the launching pad for data startups.

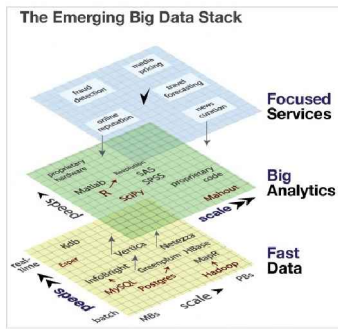
Leveraging the big data stack

As the foundational layer in the big data stack, the cloud provides the scalable persistence and compute power needed to manufacture data products.

At the middle layer of the big data stack is analytics, where features are extracted from data, and fed into classification and prediction algorithms.

Finally, at the top of the stack are services and applications. This is the level at which consumers experience a data product, whether it be a music recommendation or a traffic route prediction.

Let's take each of layers and discuss the competitive axes at each.



The competitive axes and representative technologies on the Big Data stack are illustrated here. At the bottom tier of data, free tools are shown in red (MySQL, Postgres, Hadoop), and we see how their commercial adaptations (InfoBright, Greenplum, MapR) compete principally along the axis of speed; offering faster processing and query times. Several of these players are pushing up towards the second tier of the data stack, analytics. At this layer, the primary competitive axis is scale: few offerings can address terabyte-scale data sets, and those that do are typically proprietary. Finally, at the top layer of the big data stack lies the services that touch consumers and businesses. Here, focus within a specific sector, combined with depth that reaches downward into the analytics tier, is the defining competitive advantage.

Fast data

At the base of the big data stack — where data is stored, processed, and queried — the dominant axis of competition was once scale. But as cheaper commodity disks and Hadoop have effectively addressed scalable persistence and processing, the focus of competition has shifted toward speed. The demand for faster disks has led to an explosion in interest in solid-state disk firms, such as Fusion-IO, which went public recently. And several startups, most notably MapR, are promising faster versions of Hadoop.

FusionIO and MapR represent another trend at the data layer: commercial technologies that challenge open source or commodity offerings on an efficiency basis, namely watts or CPU cycles consumed. With energy costs driving between one-third and one-half of data center operating costs, these efficiencies have a direct financial impact.

Finally, just as many large-scale, NoSQL data stores are moving from disk to SSD, others have observed that many traditional, relational databases will soon be entirely in memory. This is particularly true for applications that require repeated, fast access to a full set of data, such as building models from customer-product matrices. This brings us to the second tier of the big data stack, analytics.

Big analytics

At the second tier of the big data stack, analytics is the brains to cloud computing's brawn. Here, however, the speed is less of a challenge; given an addressable data set in memory, most statistical algorithms can yield results in seconds. The challenge is scaling these out to address large datasets, and rewriting algorithms to operate in an online, distributed manner across many machines.

Because data is heavy, and algorithms are light, one key strategy is to push code deeper to where the data lives, to minimize network IO. This often requires a tight coupling between the data storage layer and the analytics, and algorithms often need to be re-written as user-defined functions (UDFs) in a language compatible with the data layer. Greenplum, leveraging its Postgres roots, supports UDFs written in both Java and R. Following Google's BigTable, HBase is introducing coprocessors in its 0.92 release, which allows Java code to be associated with data tablets, and minimize data transfer over the network. Netezza pushes even further into hardware, embedding an array of functions into FPGAs that are physically co-located with the disks of its storage appliances.

The field of what's alternatively called business or predictive analytics is nascent, and while a range of enabling tools and platforms exist (such as R, SPSS, and SAS), most of the algorithms developed are proprietary and vertical-specific. As the ecosystem matures, one may expect to see the rise of firms selling analytical services — such as recommendation engines — that interoperate across data platforms. But in the near-term, consultancies like Accenture and McKinsey, are positioning themselves to provide big analytics via billable hours.

Outside of consulting, firms with analytical strengths push upward, surfacing focused products or services to achieve success.

Focused services

The top of the big data stack is where data products and services directly touch consumers and businesses. For data startups, these offerings more frequently take the form of a service, offered as an API rather than a bundle of bits.

BillGuard is a great example of a startup offering a focused data service. It monitors customers' credit card statements for dubious charges, and even leverages the collective behavior of users to improve its fraud predictions.

Several startups are working on algorithms that can crack the content relevance nut, including Flipboard and News.me. Klout delivers a pure data service that uses social media activity to measure online influence. My company, Metamarkets, crunches server logs to provide pricing analytics for publishers.

For data startups, data processes and algorithms define their competitive advantage. Poor predictions — whether of fraud, relevance, influence, or price — will sink a data startup, no matter how well-designed their web UI or mobile application.

Focused data services aren't limited to startups: LinkedIn's People You May Know and FourSquare's Explore feature enhance engagement of their companies' core products, but only when they correctly suggest people and places.

Democratizing big data

The axes of strategy in the big data stack show analytics to be squarely at the center. Data platform providers are pushing upwards into analytics to differentiate themselves, touting support for fast, distributed code execution close to the data. Traditional analytics players, such as SAS and SAP, are expanding their storage footprints and challenging the need

for alternative data platforms as staging areas. Finally, data startups and many established firms are creating services whose success hinges directly on proprietary analytics algorithms.

The emergence of data startups highlights the democratizing consequences of a maturing big data stack. For the first time, companies can successfully build offerings without deep infrastructure know-how and focus at a higher level, developing analytics and services. By all indications, this is a democratic force that promises to unleash a wave of innovation in the coming decade.

Data markets aren't coming: They're already here

Gnip's Jud Valeski on data resellers, end-user responsibility, and the threat of black markets.

by Julie Steele

Jud Valeski (@jvaleski) is cofounder and CEO of Gnip, a social media data provider that aggregates feeds from sites like Twitter, Facebook, Flickr, delicious, and others into one API.

Jud will be speaking at Strata next week on a panel titled "What's Mine is Yours: the Ethics of Big Data Ownership."

If you're attending Strata, you can also find out more about growing business of data marketplaces at a "Data Marketplaces" panel with Ian White of Urban Mapping, Peter Marney of Thomson Reuters, Moe Khosravy of Microsoft, and Dennis Yang of Infochimps.

My interview with Jud follows.

Why is social media data important? What can we do with it or learn from it?

Jud Valeski: Social media today is the first time a reasonably large population has communicated digitally in relative public. The ability to programmatically analyze collective conversation has never really existed. Being able to analyze the collective human consciousness has been the dream of researchers and analysts since day one.

The data itself is important because it can be analyzed to assist in disaster detection and relief. It can be analyzed for profit in an industry that has always struggled to pinpoint how and where to spend money. It can be analyzed to determine financial market viability (stock trading, for example). It can be analyzed to understand community sentiment, which has political ramifications; we all want our voices heard in order to shape public policy.

What are some of the most common or surprising queries run through Gnip?

Jud Valeski: We don't look at the queries our customers use. One pattern we have seen, however, is that there are some people who try to use the software to siphon as much data as possible out of a given publisher. "More data, more data, more data." We hear that all the time. But how our customers configure the Gnip software is up to them.

With Gnip, customers can choose the data sources they want not just by site but also by category within the site. Can you tell me more about the options for Twitter, which include Decahose, Halfhose, and Spritzer?

Jud Valeski: We tend to categorize social media sources into three buckets: Volume, Coverage, or Both. Volume streams provide a consumer with a sampled rate of volume (Decahose is 10%, for example, while a full firehose is 100% of some service's activities). Statisticians and analysts like the Volume stuff.

Coverage streams exist to provide full coverage of a certain set of things (e.g., keywords, or the User Mention Stream for Twitter). Advertisers like Coverage streams because their interests are very targeted. There are some products that fall into both categories, but Volume and Coverage tend to describe the overall view.

For Twitter in particular, we use their algorithm as described on their dev pages, adjusted for each particular volume rate desired.

Gnip is currently the only licensed reseller of the full Twitter firehose. Are there other partnerships coming up?

Jud Valeski: "Currently" is the operative word here. While we're enjoying the implied exclusivity of the current conditions, we fully expect Twitter to grow its VAR tier to ensure a more competitive marketplace.

From my perspective, Twitter enabling VARs allows them to focus on what is near and dear to their hearts — developer use cases, promoted Tweets, end users, and the display ecosystem — while enabling firms focused on the data-delivery business to distribute underlying data for

non-display use. Gnip provides stream enrichments for all of the data that flows through our software. Those enrichments include format and protocol normalization, as well as stream augmentation features such as global URL unwinding. Those value-adds make social media API integration and data leverage much easier than doing a bunch of one-off integrations yourself.

We're certainly working on other partnerships of this level of significance, but we have nothing to announce at this time.

What do you wish more people understood about data markets and/or the way large datasets can be used?

Jud Valeski: First, data is not free, and there's always someone out there that wants to buy it. As an end-user, educate yourself with how the content you create using someone else's service could ultimately be used by the service-provider.

Second, black markets are a real problem, and just because "everyone else is doing it" doesn't mean it's okay. As an example, botnet-like distributed IP address polling infrastructure is commonly used to extract more data from a publisher's service than their API usage terms allow. While perhaps fun to build and run (sometimes), these approaches clearly result in aggregated pools of publisher data that the publisher never intended to promote. Once collected, the aggregated pools of data are sold to data-hungry analytics firms. This results in end-user frustration, in that the content they produced was used in a manner that flagrantly violated the terms under which they signed up. These databases are frequently called out as infringing on privacy.

Everyone loves a good Robin Hood story, and that's how I'd characterize the overall state of data collection today.

How has real-time data changed the field of customer relationship management (CRM)?

Jud Valeski: CRM firms have a new level of awareness. They no longer rely exclusively on dated user studies. A customer service rep may know

about your social life through their dashboard the moment you are connected to them over the phone.

I ultimately see the power of understanding collective consciousness in responding to customer service issues. We haven't even scratched the surface here. Imagine if Company X reached out to you directly every time you had a problem with their product or service. Proactivity can pay huge dividends. Companies haven't tapped even 10% of the potential here, and part of that is because they're not spending enough money in the area yet.

Today, "social" is a checkbox that CRM tools attempt to check off just to keep the boss happy. Tomorrow, social data and metaphors will define the tools outright.

Have you learned anything as a social media user yourself from working on Gnip? Is there anything social media users should be more aware of?

Jud Valeski: Read the terms of service for social media services you're using before you complain about privacy policies or how and where your data is being used. Unless you are on a private network, your data is treated as public for all to use, see, sell, or buy. Don't kid yourself. Of course, this brings us all the way back around to black markets. Black markets — and publishers' generally lackadaisical response to them — cloud these waters.

If you can't make it to Strata, you can learn more about the architectural challenges of distributing social and location data across the web in real time, and how Gnip has evolved to address those challenges, in Jud's contribution to "Beautiful Data."

An iTunes model for data

Datasets as albums? Entities as singles? How an iTunes for data might work.

by Audrey Watters

As we move toward a data economy, can we take the digital content model and apply it to data acquisition and sales? That's a suggestion that Gil Elbaz (@gilelbaz), CEO and co-founder of the data platform Factual made in passing at his recent talk at Web 2.0 Expo.

Elbaz spoke about some of the hurdles that startups face with big data — not just the question of storage, but the question of access. But as he addressed the emerging data economy, Elbaz said we will likely see novel access methods and new marketplaces for data. Startups will be able to build value-added services on top of big data, rather than having to worry about gathering and storing the data themselves. “An iTunes for data,” is how he described it.

So what would it mean to apply the iTunes model to data sales and distribution? I asked Elbaz to expand on his thoughts.

What problems does an iTunes model for data solve?

Gil Elbaz: One key framework that will catalyze data sharing, licensing and consumption will be an open data marketplace. It is a place where data can be programmatically searched, licensed, accessed, and integrated directly into a consumer application. One might call it the “eBay of data” or the “iTunes of data.” iTunes might be the better metaphor because it's not just the content that is valuable, but also the convenience of the distribution channel and the ability to pay for only what you will consume.

How would an iTunes model for data address licensing and ownership?

Gil Elbaz: In the case of iTunes, in a single click I purchase a track, download it, establish licensing rights on my iPhone and up to four other authorized devices, and it's immediately integrated into my daily life.

Similarly, the deepest value will come for a marketplace that, with a single click, allows a developer to license data and have it automatically integrated into their particular application development stack. That might mean having the data instantly accessible via API, automatically replicated to a MySQL server on EC2, synchronized at Database.com, or copied to Google App Engine.

An iTunes for data could be priced from a single record/entity to a complete dataset. And it could be licensed for single use, caching allowed for 24 hours, or perpetual rights for a specific application.

What needs to happen for us to move away from "buying the whole album" to buying the data equivalent of a single?

Gil Elbaz: The marketplace will eventually facilitate competitive bidding, which will bring the price down for developers. iTunes is based on a fairly simple set-pricing model. But, in a world of multiple data vendors with commodity data, only truly unique data will command a premium price. And, of course, we'll need great search technology to find the right data or data API based on the developer's codified requirements: specified data schema, data quality bar, licensing needs, and the bid price.

Another dimension that is relevant to Factual's current model: data as a currency. Some of our most interesting partnerships are based on an open exchange of information. Partners access our data and also contribute back streams of edits and other bulk data into our ecosystem. We highly value the contributions our partners make. "Currency" is a medium of exchange and a basis for accessing other scarce resources. In a world where not everyone is yet actively looking to license data, unique data is increasingly an important medium of exchange.

This interview was edited and condensed.

Data is a currency

The trade in data is only in its infancy

by Edd Dumbill

If I talk about data marketplaces, you probably think of large resellers like Bloomberg or Thomson Reuters. Or startups like InfoChimps. What you probably don't think of is that we as consumers trade in data.

Since the advent of computers in enterprises, our interaction with business has caused us to leave a data imprint. In return for this data, we might get lower prices or some other service. The web has only accelerated this, primarily through advertising, and big data technologies are adding further fuel to this change.

When I use Facebook I'm trading my data for their service. I've entered into this commerce perhaps unwittingly, but using the same mechanism humankind has known throughout our history: trading something of mine for something of theirs.

So let's guard our privacy by all means, but recognize this is a bargain and a marketplace we enter into. Consumers will grow more sophisticated about the nature of this trade, and adopt tools to manage the data they give up.

Is this all one-way traffic? Business is certainly ahead of the consumer in the data management game, but there's a race for control on both sides. To continue the currency analogy, browsers have had "wallets" for a while, so we can keep our data in one place.

The maturity of the data currency will be signalled by personal data bank accounts, that give us the consumer control and traceability. The Locker project is a first step towards this goal, giving users a way to get their data back from disparate sites, but is one of many future models.

Who runs data banks themselves will be another point of control in the struggle for data ownership.

are the new cash-for-gold: act now to realize the neglected riches within your upper drive bay.

But the idea of “big data” as a discipline, as a conference subject, or as a business, remains in its formative years and has yet to be satisfactorily defined. This immaturity is perhaps best illustrated by the array of language employed to define big data’s merits and its associated challenges. Commentators are employing very distinct wording to make the ill-defined idea of “big data” more familiar; their metaphors fall cleanly into three categories:

- **Natural resources** (“the new oil,” “goldrush” and of course “data mining”): Highlights the singular value inherent in data, tempered by the effort required to realize its potential.
- **Natural disasters** (“data tornado,” “data deluge,” data tidal wave”): Frames data as a problem of near-biblical scale, with subtle undertones of assured disaster if proper and timely preparations are not considered.
- **Industrial devices** (“data exhaust,” “firehose,” “Industrial Revolution”): A convenient grab-bag of terminologies that usually portrays data as a mechanism created and controlled by us, but one that will prove harmful if used incorrectly.

If Strata’s Birds-of-a-Feather conference sessions are anything to go by, the idea of “big data” requires the definition and scope these metaphors attempt to provide. Over lunch you could have met with like-minded delegates to discuss big data analysis, cloud computing, Wikipedia, peer-to-peer collaboration, real-time location sharing, visualization, data philanthropy, Hadoop (natch’), data mining competitions, dev ops, data tools (but “not trivial visualizations”), Cassandra, NLP, GPU computing, or health care data. There are two takeaways here: the first is that we are still figuring out what big data is and how to think about it; the second is that any alternative is probably an improvement on “big data.”

Strata is about “making data work” — the tenor of the conference was less of a “how-to” guide, and more about defining the problem and

shaping the discussion. Big data is a massive opportunity; we are searching for its identity and the language to define it.

Data and the human-machine connection

Opera Solutions' Arnab Gupta says human plus machine always trumps human vs machine.

by Julie Steele

Arnab Gupta is the CEO of Opera Solutions, an international company offering big data analytics services. I had the chance to chat with him recently about the massive task of managing big data and how humans and machines intersect. Our interview follows.

Tell me a bit about your approach to big data analytics.

Arnab Gupta: Our company is a science-oriented company, and the core belief is that behavior — human or otherwise — can be mathematically expressed. Yes, people make irrational value judgments, but they are driven by common motivation factors, and the math expresses that.

I look at the so-called “big data phenomenon” as the instantiation of human experience. Previously, we could not quantitatively measure human experience, because the data wasn't being captured. But Twitter recently announced that they now serve 350 billion tweets a day. What we say and what we do has a physical manifestation now. Once there is a physical manifestation of a phenomenon, then it can be mathematically expressed. And if you can express it, then you can shape business ideas around it, whether that's in government or health care or business.

How do you handle rapidly increasing amounts of data?

Arnab Gupta: It's an impossible battle when you think about it. The amount of data is going to grow exponentially every day, every week, every year, so capturing it all can't be done. In the economic ecosystem there is extraordinary waste. Companies spend vast amounts of money, and the ratio of investment to insight is growing, with much more investment for similar levels of insight. This method just mathematically cannot work.

So, we don't look for data, we look for signal. What we've said is that the shortcut is a priori identifying the signals to know where the fish are swimming, instead of trying to dam the water to find out which fish are in it. We focus on the flow, not a static data capture.

What role does visualization play in the search for signal?

Arnab Gupta: Visualization is essential. People dumb it down sometimes by calling it "UI" and "dashboards," and they don't apply science to the question of how people perceive. We need understanding that feeds into the left brain through the right brain via visual metaphor. At Opera Solutions, we are increasingly trying to figure out the ways in which the mind understands and transforms the visualization of algorithms and data into insights.

If understanding is a priority, then which do you prefer: a black-box model with better predictability, or a transparent model that may be less accurate?

Arnab Gupta: People bifurcate, and think in terms of black-box machines vs. the human mind. But the question is whether you can use machine learning to feed human insight. The power lies in expressing the black box and making it transparent. You do this by stress testing it. For example, if you were looking at a model for mortgage defaults, you would say, "What happens if home prices went down by X percent, or interest rates go up by X percent?" You make your own heuristics, so that when you make a bet you understand exactly how the machine is informing your bet.

Humans can do analysis very well, but the machine does it *consistently* well; it doesn't make mistakes. What the machine lacks is the ability to consider orthogonal factors, and the creativity to consider what *could* be. The human mind fills in those gaps and enhances the power of the machine's solution.

So you advocate a partnership between the model and the data scientist?

Arnab Gupta: We often create false dichotomies for ourselves, but the truth is it's never been man vs. machine; it has always been man *plus*

machine. Increasingly, I think it's an article of faith that the machine beats the human in most large-scale problems, even chess. But though the predictive power of machines may be better on a large-scale basis, if the human mind is trained to use it powerfully, the possibilities are limitless. In the recent Jeopardy showdown with IBM's Watson, I would have had a three-way competition with Watson, a Jeopardy champion, and a *combination* of the two. Then you would have seen where the future lies.

Does this mean we need to change our approach to education, and train people to use machines differently?

Arnab Gupta: Absolutely. If you look back in time between now and the 1850s, everything in the world has changed except the classroom. But I think we are dealing with a phase-shift occurring. Like most things, the inertia of power is very hard to shift. Change can take a long time and there will be a lot of debris in the process.

One major hurdle is that the language of machine-plus-human interaction has not yet begun to be developed. It's partly a silent language, with data visualization as a significant key. The trouble is that language is so powerful that the left brain easily starts dominating, but really almost all of our critical inputs come from non-verbal signals. We have no way of creating a new form of language to describe these things yet. We are at the beginning of trying to develop this.

Another open question is: What's the skill set and the capabilities necessary for this? At Opera we have focused on the ability to teach machines how to learn. We have 150-160 people working in that area, which is probably the largest private concentration in that area outside IBM and Google. One of the reasons we are hiring all these scientists is to try to innovate at the level of core competencies and the science of comprehension.

The business outcome of that is simply practical. At the end of the day, much of what we do is prosaic; it makes money or it doesn't make money. It's a business. But the philosophical fountain from which we drink needs to be a deep one.